NASA Contractor Report 166491

Development of V/STOL Methodology Based on a Higher Order Panel Method

(NASA-CR-166491) DEVELOPMENT OF V/STOL METHODOLOGY BASED ON A HIGHER ORDER PANEL METHOD Contract Report, Mar. 1982 - Mar. 1983 (General Dynamics Corp.) 329 p HC A15/MF A01 CSCL 01A G3/02
N84-10024
Unclas
44303

I. C. Bhateley, G. A. Howell,
and H. W. Mann

CONTRACT NAS2 – 11167
FEBRUARY 1983

NASA

NASA Contractor Report 166491

Development of V/STOL Methodology Based on a Higher Order Panel Method

I. C. Bhateley, G. A. Howell,
and H. W. Mann

GENERAL DYNAMICS CORP.
FORT WORTH DIVISION
FORT WORTH, TEXAS

NASA

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

TABLE OF CONTENTS

PRECEDING PAGE BLANK NOT FILMED

TABLE OF CONTENTS (Cont'd)

LIST OF FIGURES

LIST OF FIGURES (Cont'd)

LIST OF FIGURES (Cont'd)

LIST OF FIGURES (Cont'd)

LIST OF TABLES

SUMMARY

The development of a comprehensive computational technique to predict the highly complex flowfields associated with typical V/STOL aircraft configurations has been initiated by General Dynamics. The approach being taken is to use the PAN AIR higher order panel method as the nucleus of this program and take advantage of its extensive capabilities for modeling complex configurations and applying generalized boundary conditions. A building-block approach is being used to incorporate additional capabilities into PAN AIR to account for flow phenomena generally associatd with V/STOL configurations that are not accurately treated by potential flow codes. Existing methodology and codes are being utilized whenever possible to keep development time and costs to a minimum.

The modules developed will interact with the aerodynamic code through input and output files only, thereby requiring no change to the PAN AIR code. This type of development permits easy replacement of any module or the nucleus aerodynamic code when improved versions become available. The progress made toward the development of three modules is presented in this report.

The Pre-processor Module is designed to decrease the manual data manipulation required to generate the input parameters necessary to analyze a configuration with PAN AIR. The module generates a panelling scheme that meets the critical requirements of the PAN AIR code. Graphics are also included to permit display of the configuration and rapid detection of panelling errors. The module is designed to operate interactively to allow maximum configuration flexibility.

The Engine Simulation Module is designed to take into account the mutual interference effects between the lifting and propulsion systems of the aircraft. It is currently restricted to supersonic jets at small injection angles to a subsonic freestream flow. Both blockage and entrainments effects of the plume are simulated by the module. Starting with a PAN AIR input file without any simulation of the plume effects, the module generates another file with all necessary modifications to allow the analysis of a configuration with plume effects. Preliminary validation of the module indicates that the effect of the plume on the external flowfield is correctly transmitted in all regions except in the immediate vicinity of the nozzle exit plane. The inclusion of viscous corrections to

both the plume and nozzle boattail is expected to improve the prediction in this local region.

The Viscous Flow Simulation Module is being incorporated to account for real flow effects in the predictions of V/STOL aircraft flowfields. These effects, which include decambering of lifting surface due to boundary layer buildup, trailing-edge flow separation, leading-edge bubbles, and leading-edge vortex separations, are highly prominant in most V/STOL flowfields. A two pronged approach is planned for this module. The first is to utilize an equivalent body defined from boundary layer considerations, and the second is to apply empirical corrections to the predicted data. Initial work on this module has been devoted to the task of developing the equivalent-body concept. An output file containing the flow parameters computed by PAN AIR is used to compute the boundary layer displacement thickness and separation points if any on lifting surfaces. A two-dimensional boundary layer code is applied in a strip fashion to generate this information. The RIM data base management program and several special programs written for this module are utilized to arrange the PAN AIR output in a form acceptable to the boundary layer program and vice-versa. Starting with PAN AIR input and output files the module generates a new PAN AIR input file that includes the effects of boundary layer build up and trailing-edge separation on the lifting surfaces. Check-out of this module has not been completed, but preliminary indications are that the equivalent body model chosen is not predicting the correct trends. It is not clear at this time if the problem is being caused by the equivalent body definition or by improperly computed values of pressure coefficients in the trailing-edge region.

The continued development of this type approach is highly recommended, because the only other way to get solutions for complex V/STOL flowfields is with the full Navier Stokes equations. For that we need at least a two orders of magnitude increase in computational speed over the existing Class VI computers.

1. INTRODUCTION

The need to develop tactical aircraft having V/STOL and/or STOL capability in the United States has been recognized for some time. This need has been reinforced by the role played by the British manufactured Harrier aircraft in the Falkland War. This battle scenario clearly demonstrated that aircraft with V/STOL and/or STOL capability can be effectively utilized from both sea- and land-based operations, under the most adverse conditions. The interest in incorporating these capabilities in the next generation of fighter aircraft has stimulated an interest in developing the methodology to accurately predict the aerodynamics of these configurations in low-speed flight.

The flowfields associated with V/STOL configurations operating in the hover, transition, or STOL modes are, in general, highly complex and difficult to predict. The major aerodynamic interactions encountered during the analysis of V/STOL configurations are outlined in Figure 1-1. Existing methods, in general, can only predict the force and moment coefficients for a limited class of configurations and do not generate sufficient information for a clear understanding of the flowfield, so design deficiencies may not be identified.

Government and industry have been active in the development of analytical methods for the prediction of the aerodynamic characteristics of V/STOL aircraft. The various programs have addressed a wide range of concepts including internally-flown flaps, externally-blown flaps, jet flaps, mechanical flaps with thrust vectoring, lift jets, lift fans, and vectored thrust. Most of these methods analytically model the real configuration aerodynamics with empirical modifications as required. A few of these methods are summarized below.

General Dynamics recently developed a preliminary design method (Reference 1) to estimate the thrust-induced aerodynamic forces and moments on powered-lift configurations. The method is based on the theory of Maskell and Spence (Reference 2) with appropriate empirical modifications. The incremental lift and moment coefficients predicted by this procedure for the NASA/Ames V/STOL fighter model are in good agreement with experimental data, as shown in Reference 3.

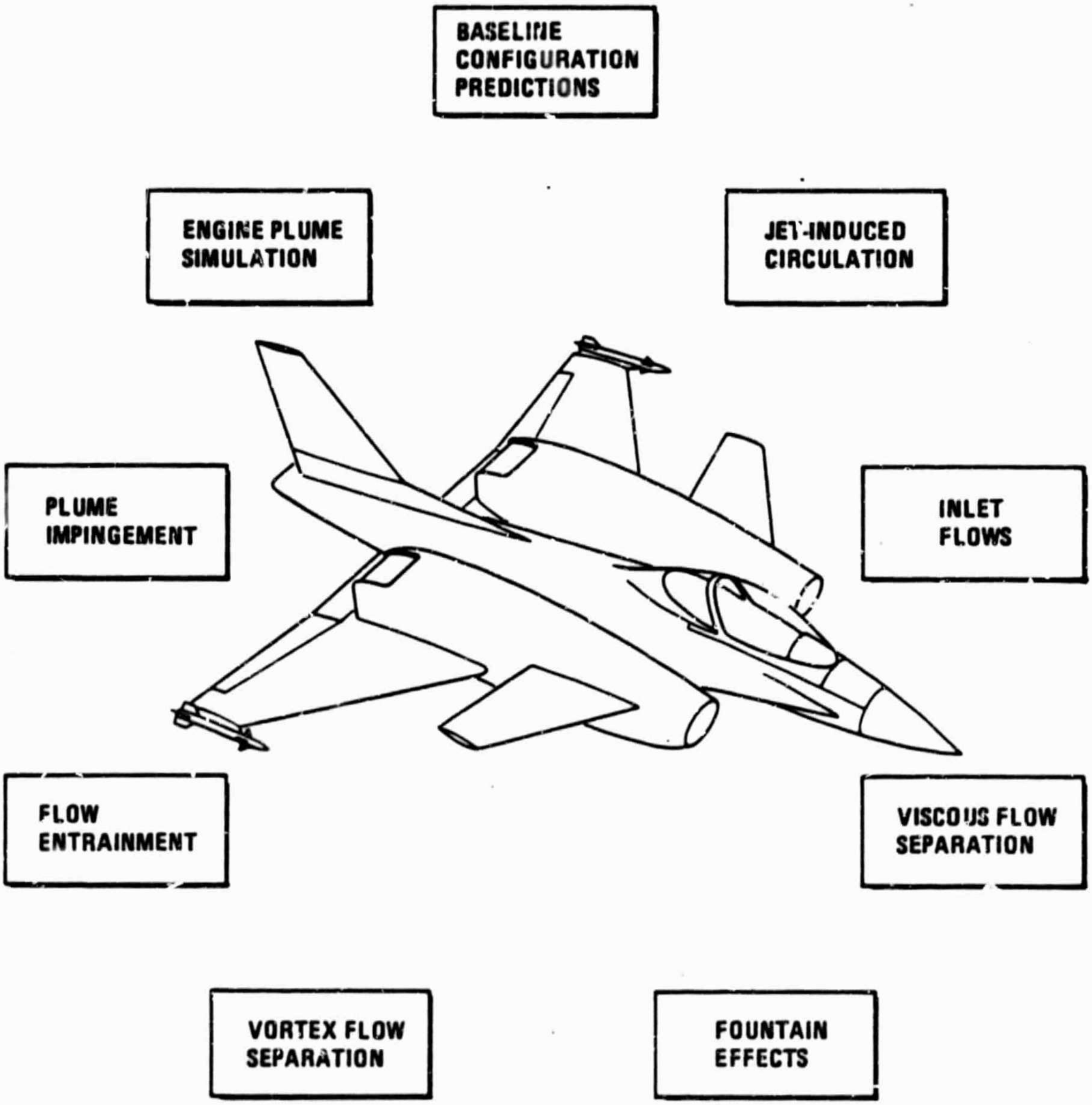


Figure 1-1 Problems Associated with the Prediction of V/STOL Aircraft Flowfields

The Convair Division of General Dynamics has developed a method for predicting incremental forces and moments due to power for V/STOL configurations (Reference 4). Various theoretical and empirical relationships were developed to account for jet-induced circulation, surface-friction (scrubbing) losses, and partial-span effects. The code computes wing lift-curve slope, induced drag, thrust recovery, flap pitching-moment increment, downwash, and lateral/directional stability derivatives. The jet-flap lifting-line theory of Kerney (Reference 5) is used to estimate wing lift-curve slope with power on.

Rockwell International has developed a method to analyze configurations with externally blown flap systems (Reference 6). The lift curve slope in this procedure is based on the two-dimensional, thin-airfoil, jet-flap theory of Spence (References 7 and 8) with the Maskell and Spence aspect-ratio correction employed to account for three-dimensional effects. The method for the maximum lift coefficient is entirely empirical. In addition to forces and moments, predictions are included for lift curve slope, maximum lift coefficient, induced drag, thrust recovery, downwash, asymmetric engine operation, and static longitudinal and lateral/directional stability derivatives.

A Boeing Company method (Reference 9) is designed to predict forces and moments on aircraft that employ mechanical flap/vectored thrust combinations or internally-blown flap systems. The mechanical flap system method is basically empirical and uses an equivalent jet-velocity ratio as a similarity parameter for the prediction of the power-induced incremental lift, drag, and moment coefficients. The internally-blown flap method employs an analysis based on a jet-flap lifting-line theory concerned principally with the influence of non-planar trailing vortices on the wing aerodynamics.

Northrop Aircraft has developed a specialized aerodynamic force and moment prediction method (Reference 10) for V/STOL aircraft that employ a lift jet, lift fan, or vectored thrust in the hover and transition flight regimes. This method uses a building block approach in which the basic flow problems are solved independently. For example, unpowered aerodynamics and power-induced effects are added together to obtain the total aerodynamics. Various components of the aircraft are also treated individually and added together to obtain total aircraft aerodynamics. The non-linear wing aerodynamic characteristics are analyzed by a double lifting-line theory, and the body aerodynamics are accounted for by modified slender-body theory. A significant feature of the Northrop method is its ability to compute the aerodynamics of transverse flow such as that emitted

from a lift jet, lift fan, or vectored-thrust nozzle. The method takes into account entrainment into the jet due to viscous mixing and distortion of the jet from a circular to a kidney-shaped cross-section, as the jet develops downstream. The jet is assumed to be an elliptical-shaped solid body for purposes of drag estimation. Conformal mapping techniques are used to analyze the interference effects of jets on the body and wing components.

The Douglas Aircraft Company has developed a method based on a linearized lifting-surface theory for jet-flapped wings, which Douglas identifies as the Elementary Vortex Distribution (EVD) method (References 11, 12, and 13). This method has the capability of predicting surface pressures and empirically corrected forces and moments. It is specifically geared to the design and analysis of STOL transport aircraft employing internally-blown jet flaps, externally-blown jet flaps, and mechanical flap systems with vectored thrust. The EVD method is capable of handling zero-thickness wings of arbitrary planform with arbitrary spanwise distributions of twist, flap chord, jet-blowing momentum, and jet-deflection angle. It represents one of the most general and exact solutions available for jet-flapped finite-span wings.

The NASA/Langley Research Center and the University of Kansas have jointly developed a linear inviscid subsonic compressible flow theory to treat the aerodynamic interaction between the wing and an inviscid upper-surface-blowing thick jet. This method provides predictions for numerous jet parameters including jet temperature, Mach number, aspect ratio, exit area, deflection angle, and proximity to the wing. Forces and moments are determined by integrating the theoretical pressures.

The Vought Corporation recently developed a code (Reference 14) that integrated three separate plume modeling methods and an inlet method with a potential flow code. It allows the user to select which of the plume methods will be used in the analysis. This is a sound approach to modeling aerodynamic/propulsive interactions, but the plume modeling methods are not general enough to handle several practical problems that are often encountered in the analysis of modern fighter aircraft.

The analytical methods discussed above and others, too numerous to mention, provide many theoretical, empirical, and semi-empirical methods to estimate the aerodynamic characteristics of V/STOL aircraft. However, each method is limited in the classes of vehicles for which it is specifically designed and often to a very limited range

of flight conditions. These restrictions are caused primarily by the empiricism normally incorporated into theoretical procedures to obtain results for the given class of configurations under study.

Exact solutions for the flowfields associated with arbitrary V/STOL configurations require solution of the Navier-Stokes equations. This is clearly beyond the current capabilities of the available computer hardware. A better near-term approach for modeling free air transition and forward flight flow phenomena is illustrated in Table 1. This approach is based on a modern panel method to predict the aerodynamics of the baseline configuration extended and modified to account for propulsion effects and real flow effects.

Table 1 V/STOL AERODYNAMIC PREDICTIONS APPROACH

Power-Off Characteristics	Baseline Configuration o Forces & Moments o Pressures
Powered Flight Effects	Inlet Flow o Spillage o Ram Drag o Interference Exhaust Flow o Plume Shape o Entrainment o Impingement o Supercirculation o Fountain Effects o Reaction Jets
Real Flow Effects	Viscous Flow o Separation o Tip Vortex o L.E. Vortex o T.E. Vortex Rollup

The success of a panel-type approach hinges on several factors. Most important is the selection of the baseline analysis code, which must accurately predict the aerodynamics of complex configurations and also, allow application of general boundary conditions that will be used to simulate power and viscous effects. A major objective is to devise a method for the analysis of configurations in which the propulsive system is closely coupled with the lifting surfaces. The interactions between the aerodynamic and propulsive flowfields can be quite strong on some configurations, and therefore, need to be predictable so that designers can take advantage of these potentially favorable effects. Another objective is to analyze flowfields where viscous phenomena dominate. For instance, separated and vortex flows, which are significant features of the flowfield of V/STOL aircraft operating at high angles of attack. General Dynamics has initiated the development of a V/STOL aerodynamic prediction technique in which a building-block approach is being used to enhance the capabilities of a modern aerodynamic potential-flow code.

2. TECHNICAL APPROACH

General Dynamic's plan for a V/STOL aerodynamic code, includes the development of distinct modules that will be mated to a potential-flow aerodynamic code. Special methods to handle the flow phenomena not adequately modeled by potential-flow codes, such as viscous effects, will be integrated into the overall code in a modular manner. The following subsections discuss the overall aspects of the code.

2.1. AERODYNAMIC PREDICTION CODE

General Dynamics has selected the PAN AIR code as the aerodynamic flow prediction method. This code was developed by The Boeing Military Airplane Company under the joint sponsorship of NASA's/Ames and Langley Research Centers, the Air Force's Aeronautical Systems Division and Wright Aeronautical Laboratory, and the Naval Coastal System Center. This higher order panel method has been selected because it provides more flexiblility than any other panel method. Similar codes developed by Woodward and Hess, although excellent codes, just do not match the overall capabilities of PAN AIR. A brief description of the theoretical method used within PAN AIR is presented in this report and the reader is referred to the code documentation (References 15 through 18) for a detailed discussion.

PAN AIR solves the three-dimensional boundary value problem governed by the second order partial differential equation usually referred to as the Prandtl-Glauret equation.

$$(1-M_\infty^2)\phi_{xx} + \phi_{yy} + \phi_{zz} = 0$$

The above equation is valid for the analysis of incompressible and linear compressible subsonic and supersonic flow of arbitrary fluids over arbitrary configurations. Mixed subsonic and supersonic flows and discontinuities in the flow cannot be adequately simulated by this differential equation. Similar to other panel methods, PAN AIR divides the configuration into segments, each of which is further divided into quadrilateral panels. Here the similarity between PAN AIR and most other panel methods ends. PAN AIR is a higher order panel method because it does not use a constant singularity strength over each panel but allows linear source strength and quadratic doublet strength variation on each panel. To further improve accuracy, PAN AIR enforces continuity of doublet

strength over the entire configuration and also maintains continuity of the geometric surface by subdividing each quadrilateral panel into several piecewise flat panels. By using these refinements, the solutions generated by PAN AIR become almost independent of the panelling scheme.

The PAN AIR code is very flexible in the type of boundary conditions that can be used to define the singularity strengths on the panels. A general form of the boundary condition is incorporated into the code. This allows the user to select boundary conditions which simulate impermeable surfaces, surfaces with specified normal or tangential velocities etc. User selected boundary conditions can be applied over different parts of the configuration since the number of unknown singularity strengths and boundary conditions are matched over each network.

The capabilities of PAN AIR (taken from Reference 10, include the ability to

- Analyze completely arbitrary configurations in subsonic flow and nearly arbitrary configurations in supersonic flow.
- Analyze either unsymmetric configurations or configurations with one or two planes of symmetry.
- Analyze configurations in either unsymmetric or symmetric flight conditions, including ground effect conditions.
- Analyze or design both geometrically thick configurations and thin configurations, such as a camber surface representation of a thin wing.
- Analyze configurations either (in an exact sense) with boundary conditions applied on the configuration surface or (in a linearized sense) with appropriate boundary conditions applied to an approximation to the configuration surface.
- Analyze control surface deflections either (in an exact sense) by geometric deflection of the appropriate networks or (in a linearized sense) by imposing suitable boundary conditions on an approximation to the deflected control surface.

- Design the location of surfaces, including wakes, by the non-iterative design capability.

- Superimpose incremental velocity components onto the freestream either in a global sense, for example, additional velocity components to simulate a finite roll rate, or on a local basis, for example, to simulate different angles of attack for different networks or to simulate the effects of a slipstream or line vortex.

- Calculate pressure coefficients and force moment coefficients by several pressure coefficient formulas (isentropic, linearized, second-order, reduced second-order, and slender body).

- Calculate velocity components and pressure coefficients both at standard points and at user-designated arbitrary points on the configuration surface (flow quantities in the external flow field can be computed by using panels with zero singularity strength).

- Calculate the force and moment coefficients on individual panels, columns of panels, and networks with the options of using user-specified reference dimensions and moment axes of either individual networks or the total configuration.

- Include or exclude the force and moment contributions of individual networks in the calculation of the force and moment coefficients of the total configuration.

- Calculate force and moment coefficients in the reference axis system, in the stability axis system, in the wind axis system, and in a user-specified body axis system.

- Calculate leading- and side-edge forces and moments due to singularity of the leading- and side-edge force distributions for thin configurations, and to include these calculations in the total configuration force and moment coefficients.

PAN AIR code has been developed in a modular fashion using a structured software development approach. This, together with a liberal distribution of comment cards throughout the code, allow for rapid upgrading of the code where necessary. The various

modules of the code communicate with each other only through common data bases so that changes made in one module affect the other modules in a clearly identifiable manner.

2.2 V/STOL ANALYSIS MODULES

The function of each module developed in this study is briefly discussed below and then discussed in detail in the following sections of this report.

Preprocessor Module - Wide acceptance of a computer code requires that it be user friendly. This in general implies that an engineer can understand the code, analyze a problem, and interpret the computed results within a reasonable time. PAN AIR is a very general code that requires detailed input for many parameters, some of which can easily be foreset for V/STOL type problems. The basic goal for the preprocessor is to make PAN AIR easier to use for V/STOL analysis by reducing the number of input parameters, developing an automatic panel generation and modification scheme, and including interactive graphic displays of the panelling arrangement.

Engine Simulation Module - In general, the engine placement in V/STOL configurations is such that the interactions of the inlet and nozzle flows with the aircraft external flowfield should not be ignored. A method to model a plume as a permeable body with specified inflow velocities to simulate entrainment effects has been developed in this study and is discussed in a later section of this report. A similar technique will be necessary to simulate inlets by either specifying inlet velocities at the inlet face or by modeling the capture tube as a surface.

Viscous Effects Module - V/STOL aircraft often operate at high lift coefficients and flow inclinations and hence, incur significant flow separation on the lifting surfaces and bodies. These areas of separated flow often lead to strong vortex formulations. Simulation of these effects is a very difficult problem. The development of this module has been initiated. Methods have been incorporated to simulate the effect of boundary layer build up and trailing edge separation on lifting surfaces using 2-D boundary layer methods in a strip fashion. To include boundary layer effects on a body-and nacelle-type surface, it will be necessary to utilize three-dimensinal boundary layer codes. Simulation of vortical flows may have to be handled by empirical corrections to the computed results.

3. VALIDATION OF THE AERODYNAMIC CODE

The mathematical concepts incorporated in the PAN AIR code have been substantiated in numerous studies utilizing an earlier version of the code referred to as the PAN AIR Pilot Code (Reference 19). On the other hand, the PAN AIR code is still in the final stages of development and checkout. Since the code was not generally available to industry until 1981, it has not yet received the extended useage that is required to identify all the programming errors that invariably occur during the development of such complex computer programs. Several versions of the code have been released since that time, and the success in using them has been mixed. The early work at General Dynamics was directed towards the analysis of a complex VATOL fighter model. It was found that complex configurations sometimes caused problems, but that these problems often could not be recreated on simplier geometries when attempts were made to isolate their sources. In other words, the code would perform correctly with less complex configurations. The early results of these studies are presented in detail in Reference 20.

In the study reported in this document, attempts were continued to obtain an improved flowfield prediction for the Ames/GD V/STOL Fighter model. Efforts were also initiated to compute the aerodynamic characteristics of the Ames/Vought VATOL model. The latest version of the PAN AIR code, released in January 1983, was utilized in this study. Preliminary indications are that some of the errors in the code, though not all, have been corrected. The following subsections discuss the power-off predictions for these two V/STOL models performed with the latest version of the code.

3.1 AMES/GD V/STOL FIGHTER MODEL

The Ames/GD V/STOL Fighter model is approximately a 75% scale of an operational fighter designed to perform an air-superiority mission. During tests in the Ames 40 by 80-Foot Wind Tunnel, the model was powered by two turbojet engines. A general arrangement drawing of this model is shown in Figure 3-1. The locations of the wing and canard chordwise pressure instrumentation are shown in Figure 3-2. The pressure and force data that have been obtained during tests of this model are presented in References 21 through 25. Several variations of the panelling arrangement for the Ames/GD V/STOL model have been used since the inception of this study, the latest of which is shown in Figure 3-3.

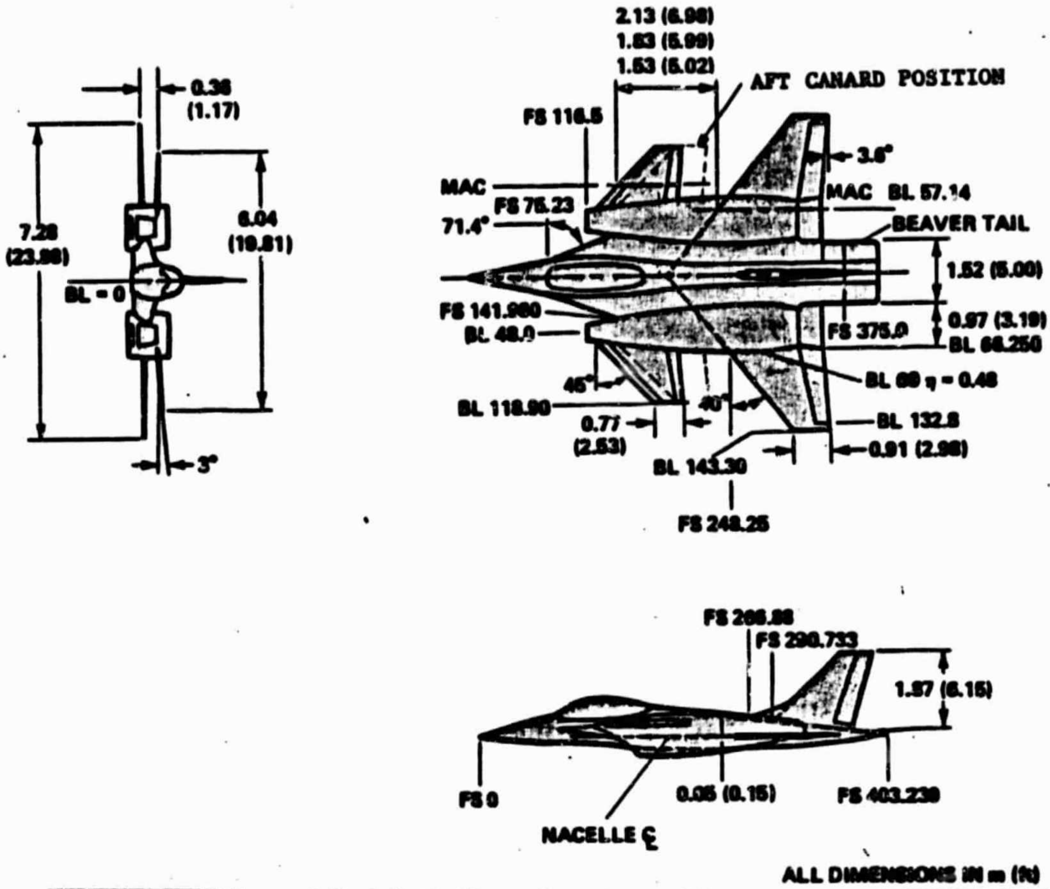


Figure 3-1 General Arrangement Drawing of the Ames/GD V/STOL Fighter Model

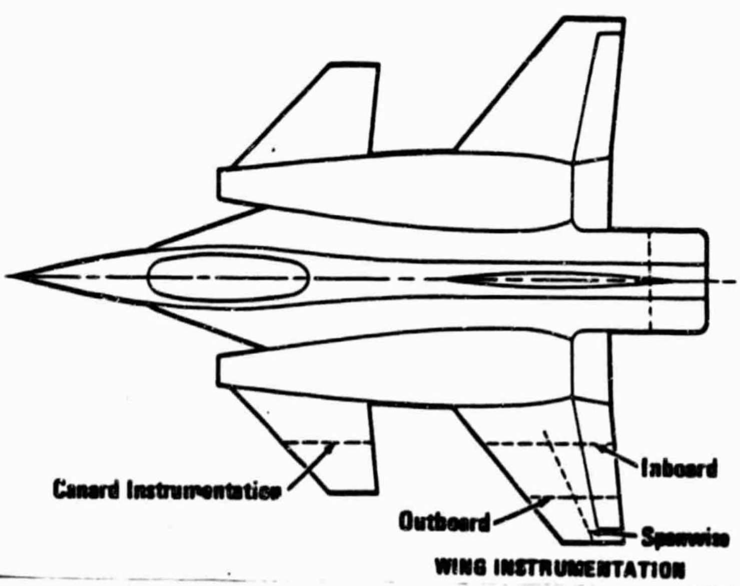


Figure 3-2 Pressure Instrumentation on the Ames/GD Model

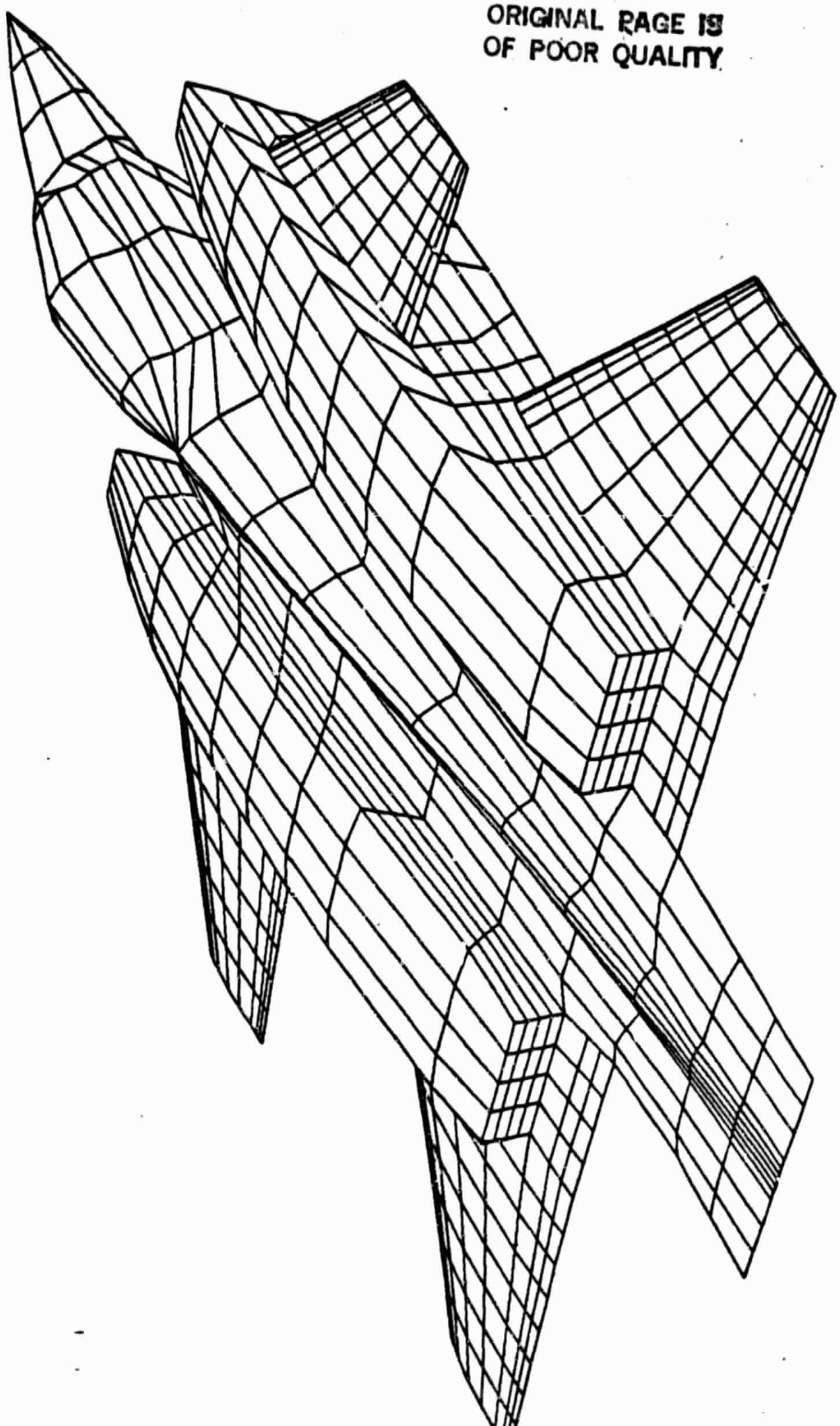


Figure 3-3 Panelling Arrangement for the Ames/GD Model

The PAN AIR prediction for chordwise pressures on the wing and canard are compared with test data in Figures 3-4, 3-5, and 3-6. These predictions are not substantially different from those reported earlier in Reference 20. The predictions compare reasonably well with the data, except in regions where flow separation effects have been identified in the test data. The types of pressure formations that result from the presence of a leading-edge vortex on this model are discussed in detail in References 23 and 25. One questionable aspect of the panelling arrangement for wing/canard configurations is the proper placement of the canard wake over the wing. Especially at higher angles of attack, the assumed straight-back trailing position of the canard wake may not be in the best location relative to the wing. An investigation of this effect was not undertaken during this study.

A test-to-theory comparison of the lift curve of this model (Figure 3-7) shows the prediction to be lower than the test data. These results are also similar to those presented in Reference 20. There are some reservations about both the test data and the computed results presented in this figure. The test data shows a very high value of lift coefficient at zero alpha, which may be due to the large amount of body camber. The possibility of an error in the experimental lift coefficient must also be considered.

The PAN AIR prediction was obtained from an analysis that utilized some artifices to obtain a solution. First, the latest version of the code would not execute with gap-filling panels. To eliminate them, minor modificatons to the geometry were required that should not have caused any significant changes in the computed results. Second, a partial-edge, mid-network abutment on the side of the nacelle, downstream of the canard, had to be specified in the input to eliminate warning messages about empty-space abutments. These abutments had exact corner-point matching and should not have required this specification, thus pointing to a possible error in the PAN AIR code. Finally, some of the wake networks near the nozzle exhaust had extremely low values of pressure coefficient, as shown in Figure 3-8.

The artifices used to obtain a PAN AIR solution seem minor and would not be expected to have significant impacts on the computed lift coefficients. However, since differences exists between the test and predictions, both should be reexamined to resolve the discrepancy. The Ames/GD V/STOL model is an extremely complex arrangement to analyze with a panel method. It is noted that PAN AIR predictions for the lift coefficients of the F-16XL and the Ames/Vought models that have been performed at General Dynamics have compared well with experimental data.

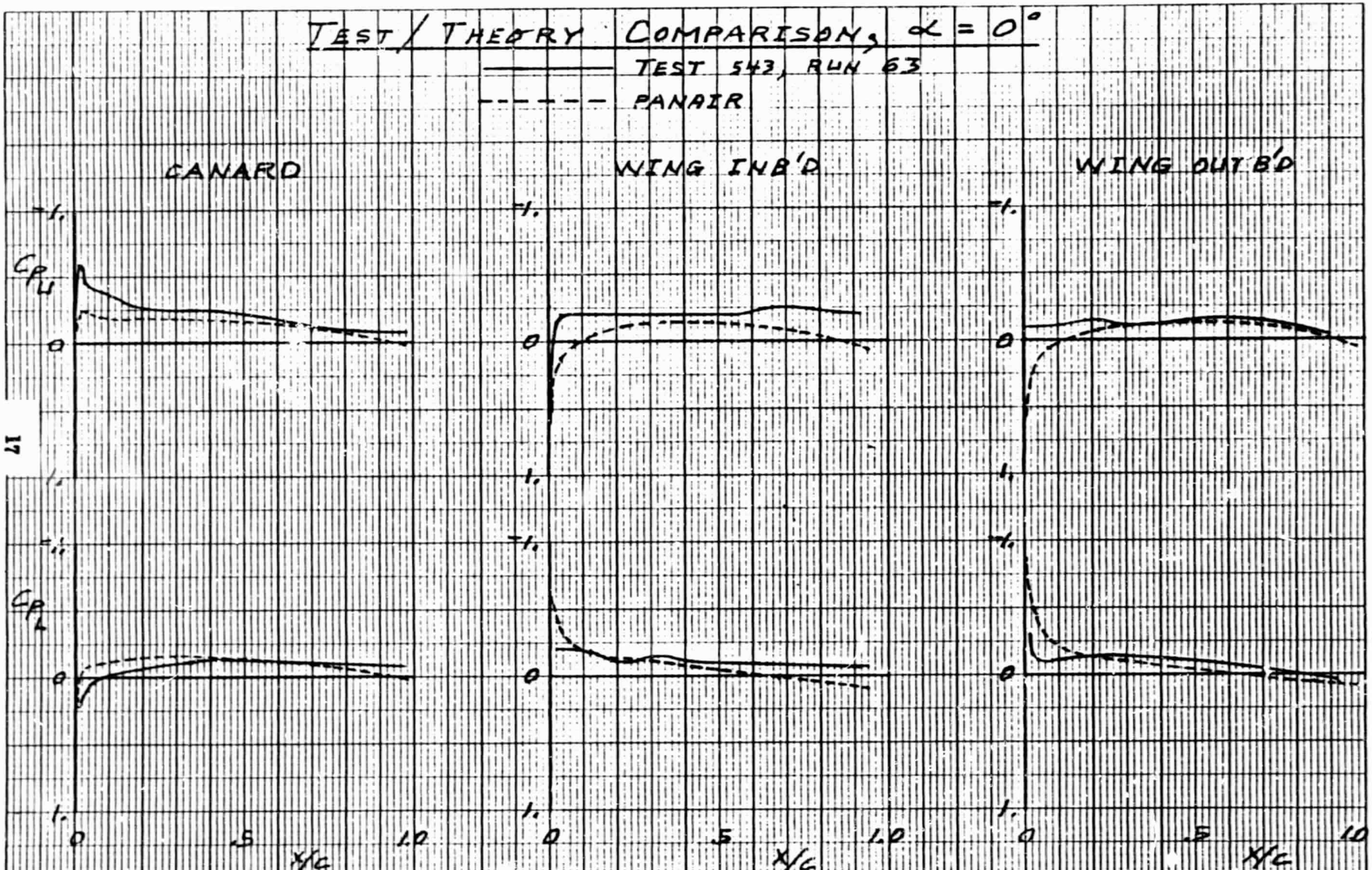


Figure 3-4 Test-to-Theory Comparisons of Pressure Data for the Ames/GD Model, Alpha = 0 Deg.

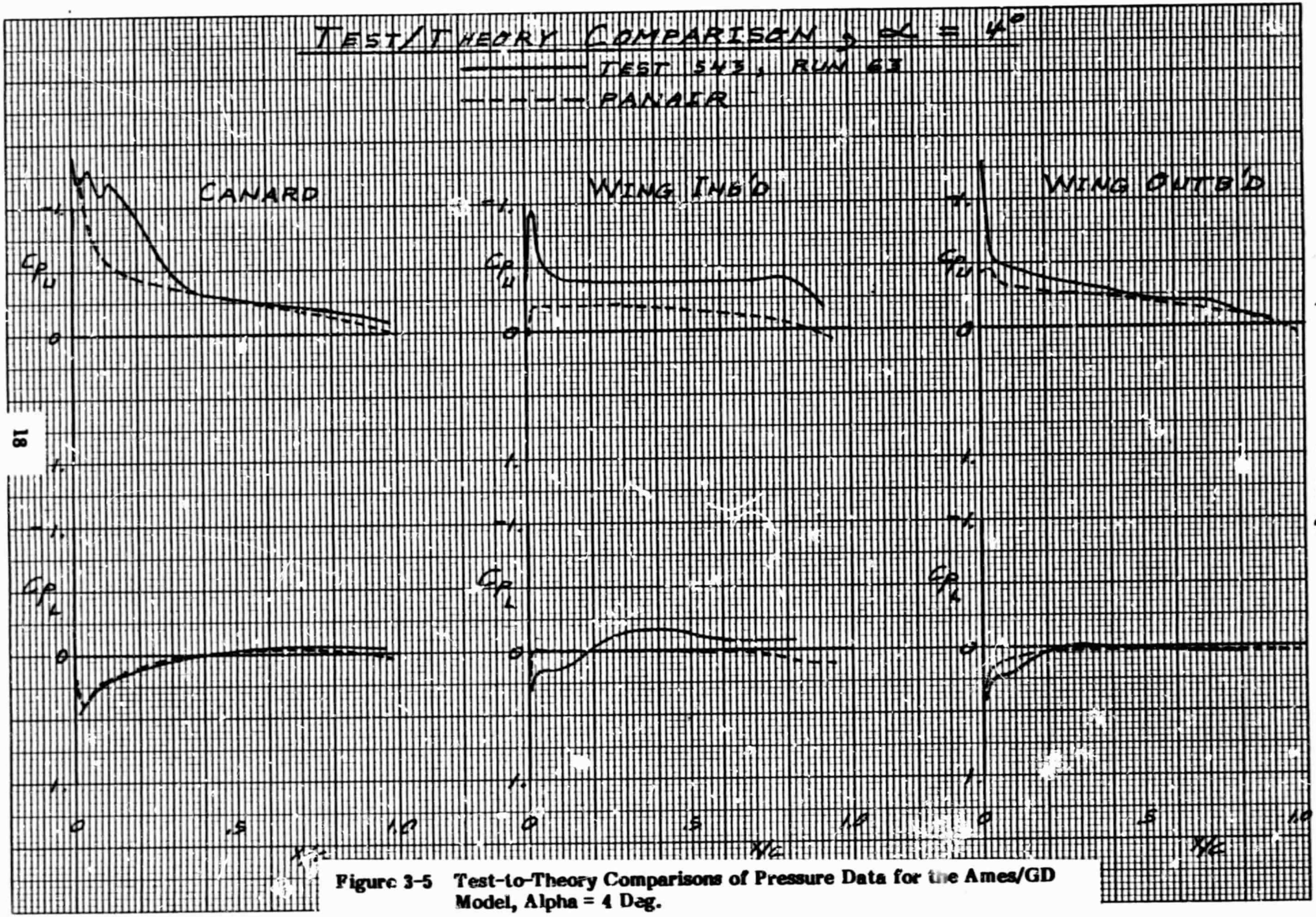


Figure 3-5 Test-to-Theory Comparisons of Pressure Data for the Ames/GD Model, Alpha = 4 Deg.

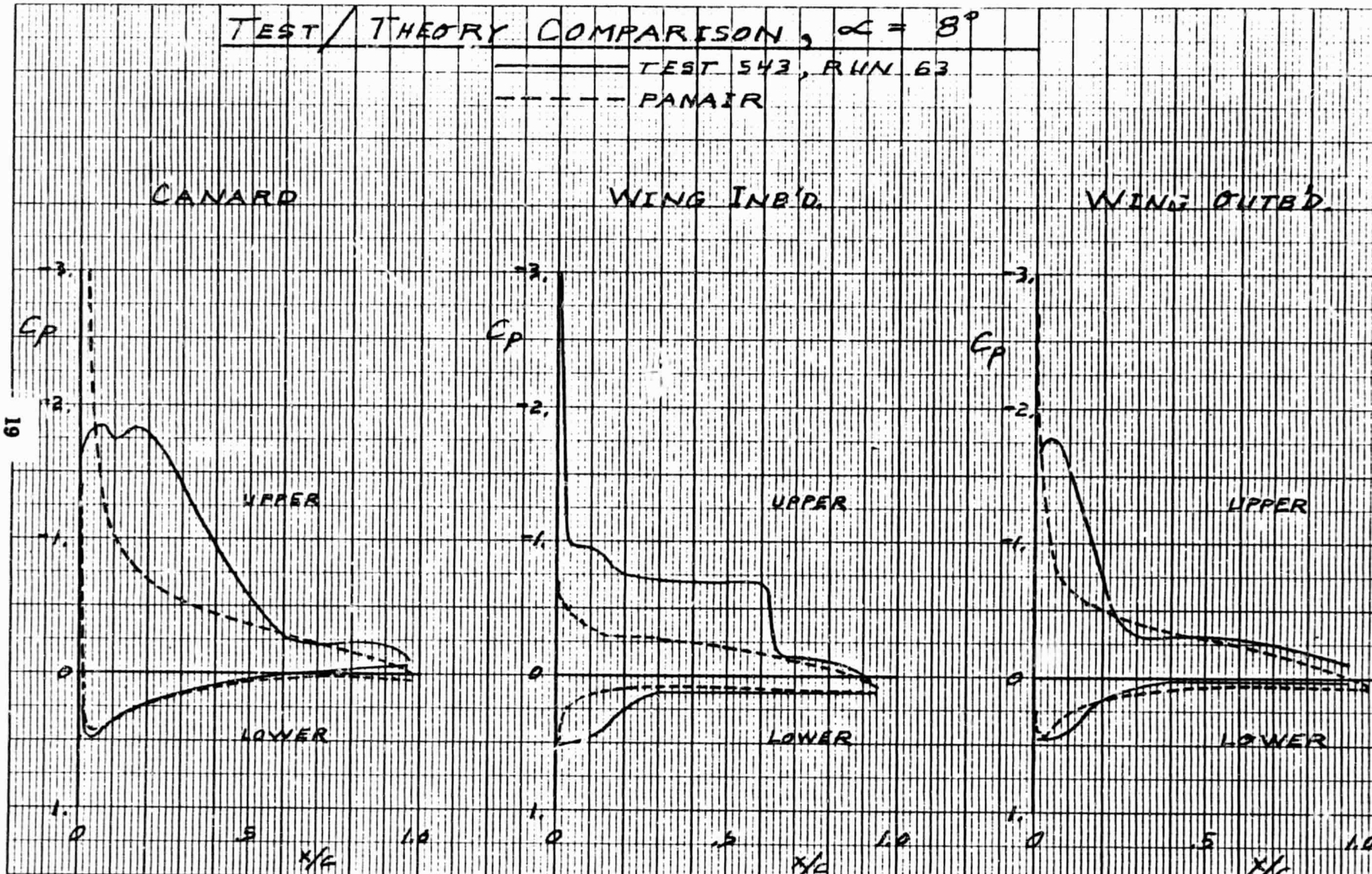


Figure 3-6 Test-to-Theory Comparisons of Pressure Data for the Ames/GD Model, Alpha = 8 Deg.

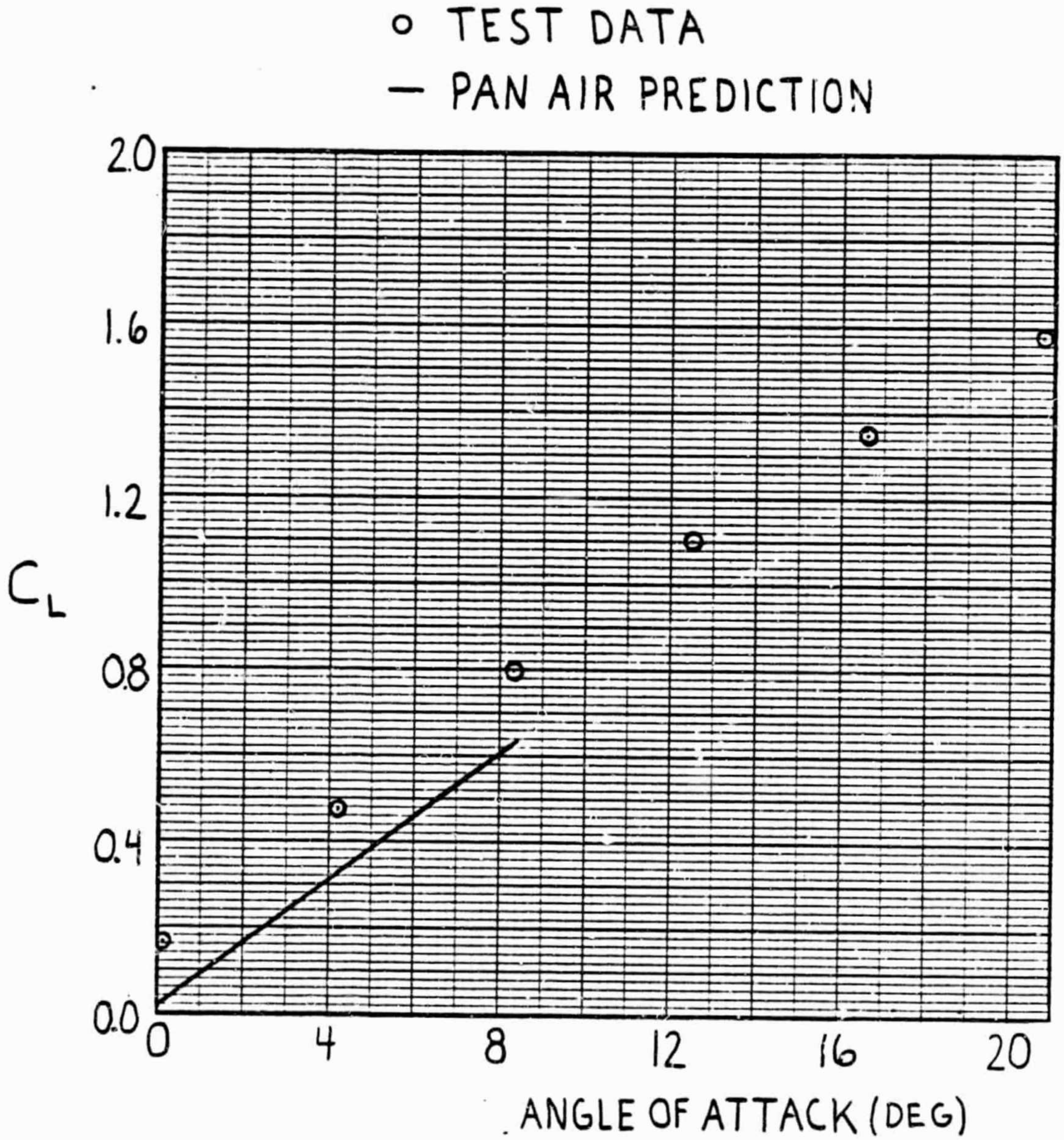


Figure 3-7 Test-to-Theory Comparison of the Lift Curve for the Ames/GD Model

ORIGINAL PAGE IS
OF POOR QUALITY

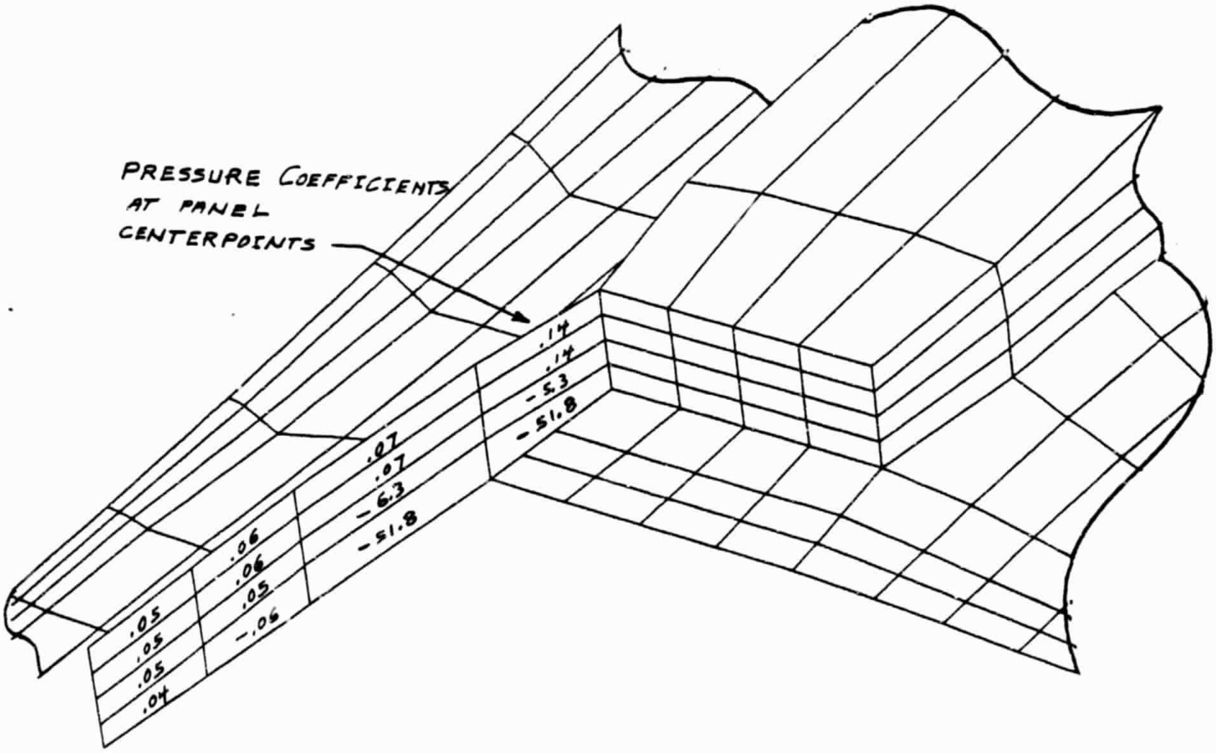


Figure 3-8 Pressure Coefficients Near the Nozzle of the Ames/GD Model

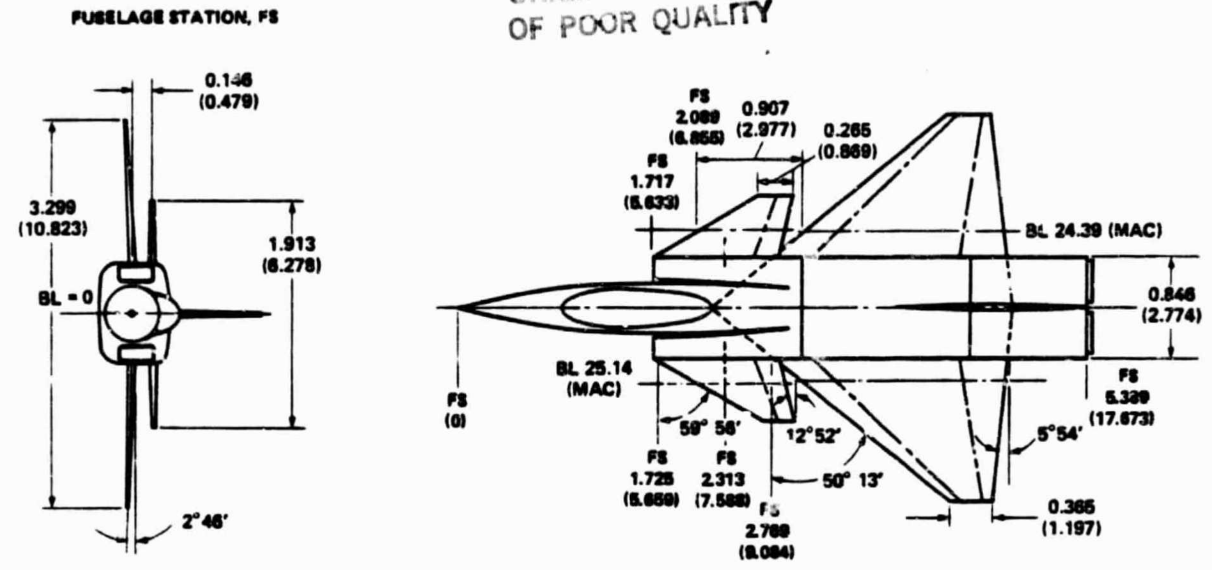
3.2 AMES/VOUGHT VATOL MODEL

The Ames/Vought model is a 0.4-scale powered VATOL model designed for testing at angles of attack to 90 deg and greater. A general arrangement drawing for this model and the locations of the pressure instrumentation are shown in Figures 3-9 and 3-10. The model was tested with power off and also with a high RPM fan to simulate inlet flow. Wind tunnel test data for this model are presented in References 26 and 27. The original panelling arrangement for this model was obtained from NASA but was later enhanced through the use of the Preprocessor Module, as discussed in Section 4. The enhanced panelling arrangement is shown in Figure 3-11.

Test-to-theory comparison of the chordwise pressures on the canard surfaces are presented in Figures 3-12, 3-13, and 3-14 for three spanwise stations. For each location, data are presented at angles of attack of 4, 8, and 20 deg. Experimental data at 8 and 20 deg were available only with the engine fan operating at 10,000 RPM. The effect of the inlet flow at these angles of attack is assumed to be small. Experimental data indicate the formation of a leading-edge vortex, similar to that of the fighter model and trailing-edge separation characteristics, at high angles of attack. These factors account for the major differences between experimental and predicted values.

Test-to-theory comparisons of the wing pressure data are shown in Figures 3-15, 3-16, and 3-17. They are similar to the comparison for the canard. Again the effects of trailing-edge separation and vortical flow have considerable influence on the test data, which explain some of the large differences between test and theory.

The computed lift curve, Figure 3-18, compares well with the test data up to an angle of attack of approximately 12 degrees. Above this angle, additional lift, presumably generated by the leading-edge vortex, causes the test data to be slightly higher than the prediction.



WING:	
REFERENCE AREA, m² (ft²)	4.752 (51.15)
ASPECT RATIO	2.29
TAPER RATIO	0.145
AIRFOIL SECTION	NACA 65A004
MEAN AERO CHORD, m (ft)	1.708 (5.605)
CANARD:	
AREA, m² (ft²)	0.702 (7.56)
CANARD AREA/WING AREA	0.148
ASPECT RATIO	1.60
TAPER RATIO	0.250
AIRFOIL SECTION	NACA 65A004
MEAN AERO CHORD, m (ft)	0.742 (2.434)

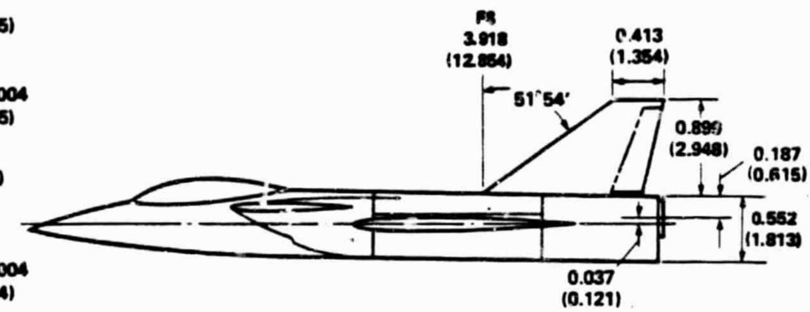


Figure 3-9 **General Arrangement Drawing of the Ames/Vought VATOL Fighter Model**

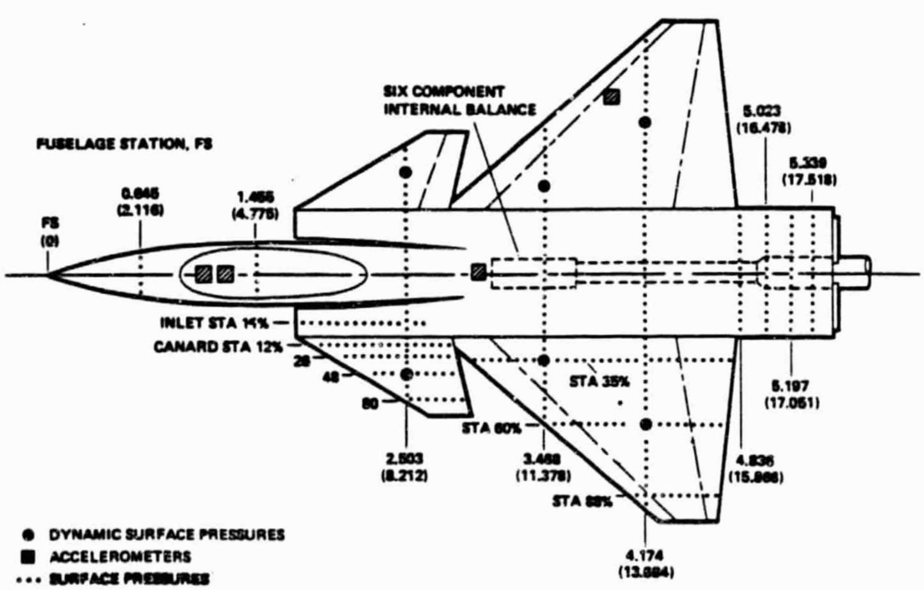


Figure 3-10 Pressure Instrumentation on the Ames/Vought Model

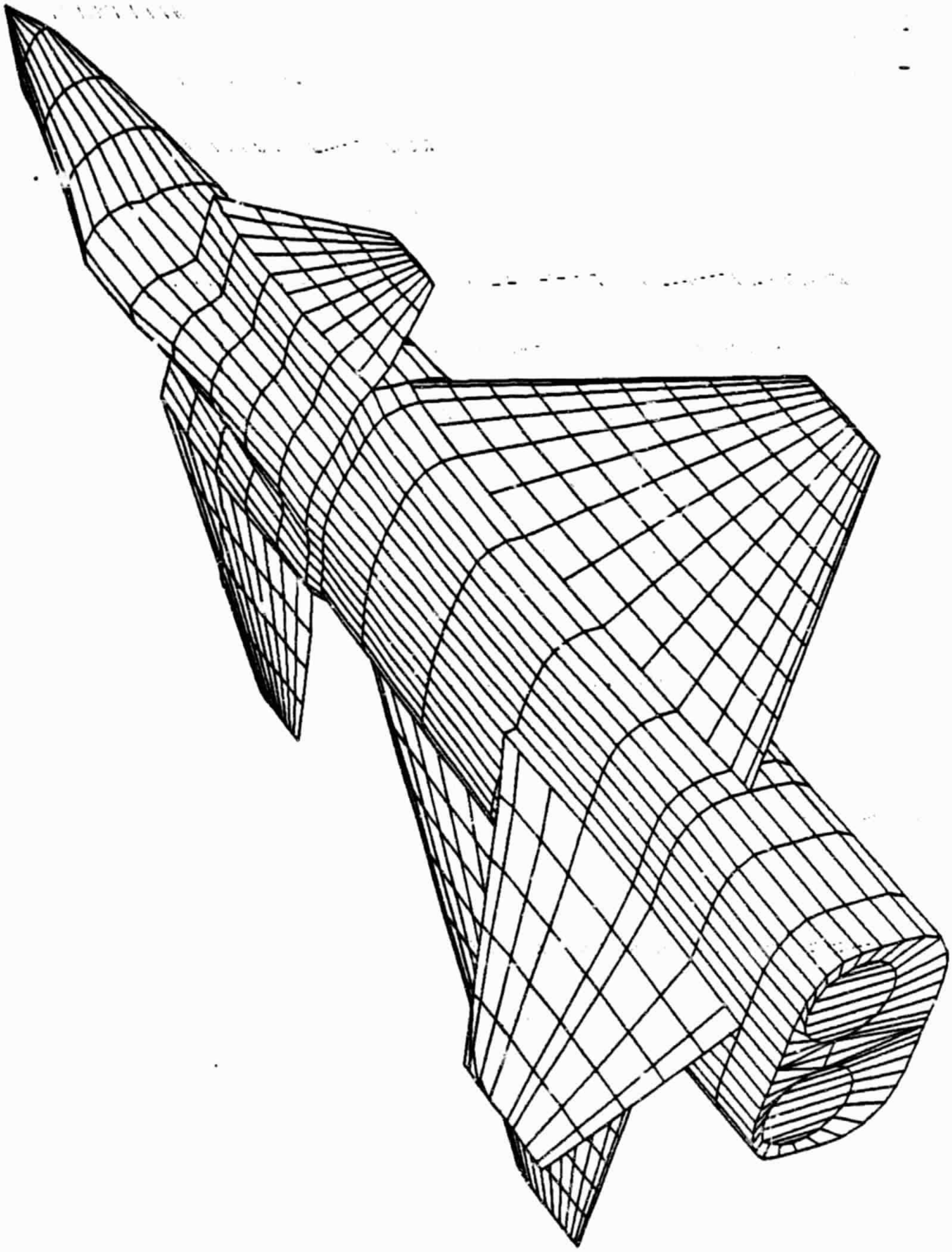


Figure 3-11 Panelling Arrangement for the Ames/Vought Model

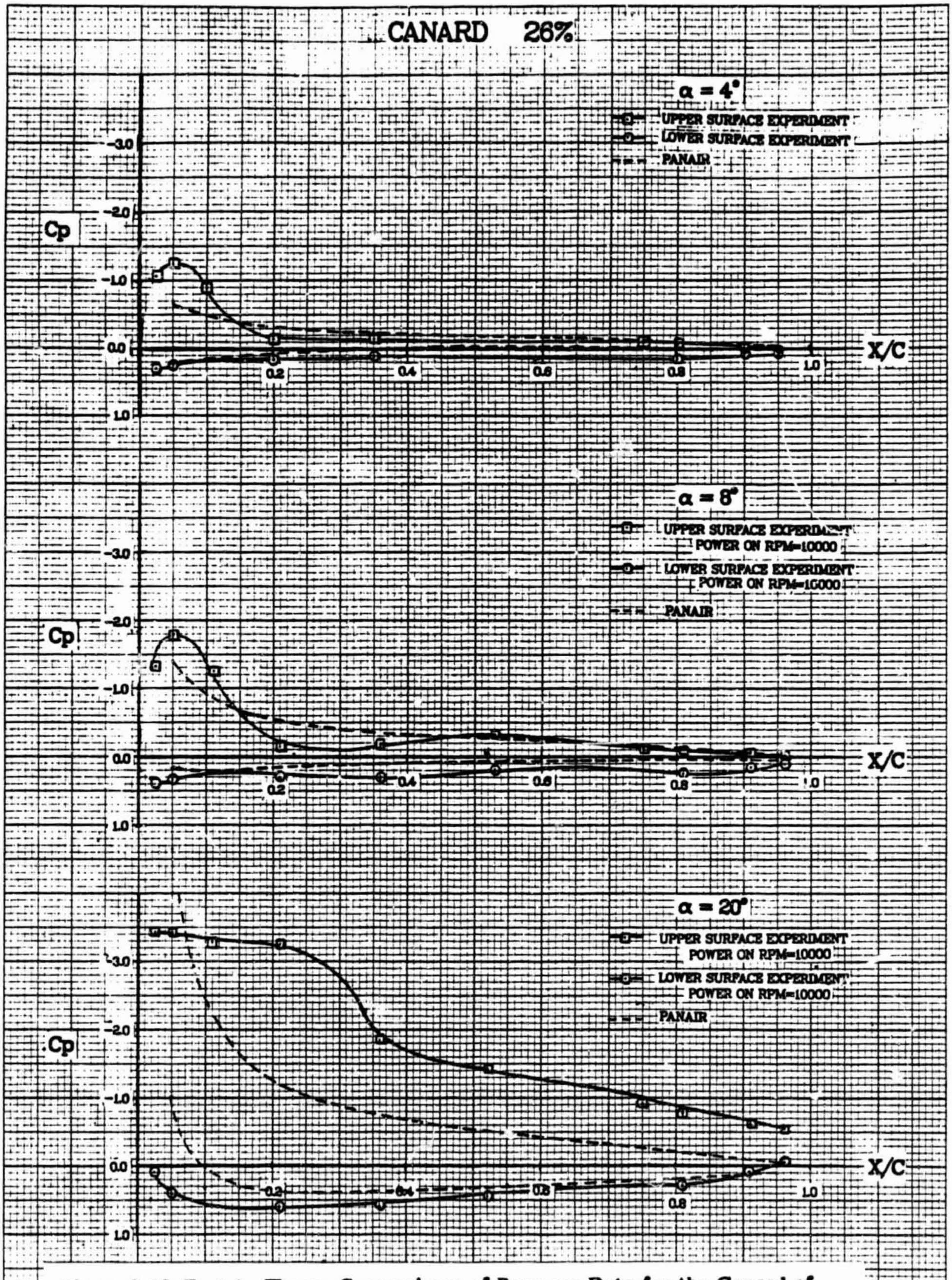


Figure 3-12 Test-to-Theory Comparisons of Pressure Data for the Canard of the Ames/Vought Model, 26% Semispan

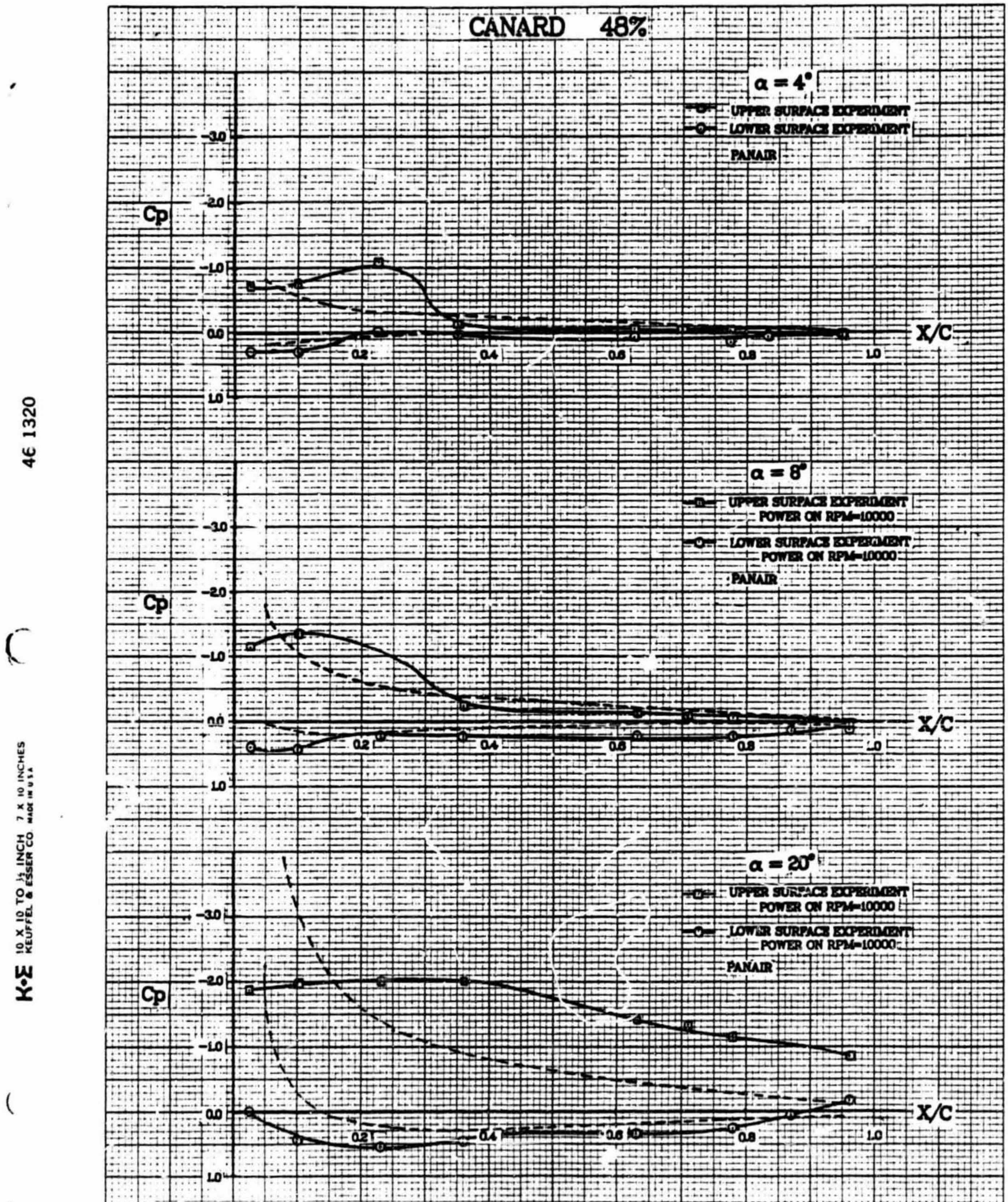


Figure 3-13 Test-to-Theory Comparisons of Pressure Data for the Canard of the Ames/Vought Model, 48% Semispan

ORIGINAL PAGE IS
OF POOR QUALITY

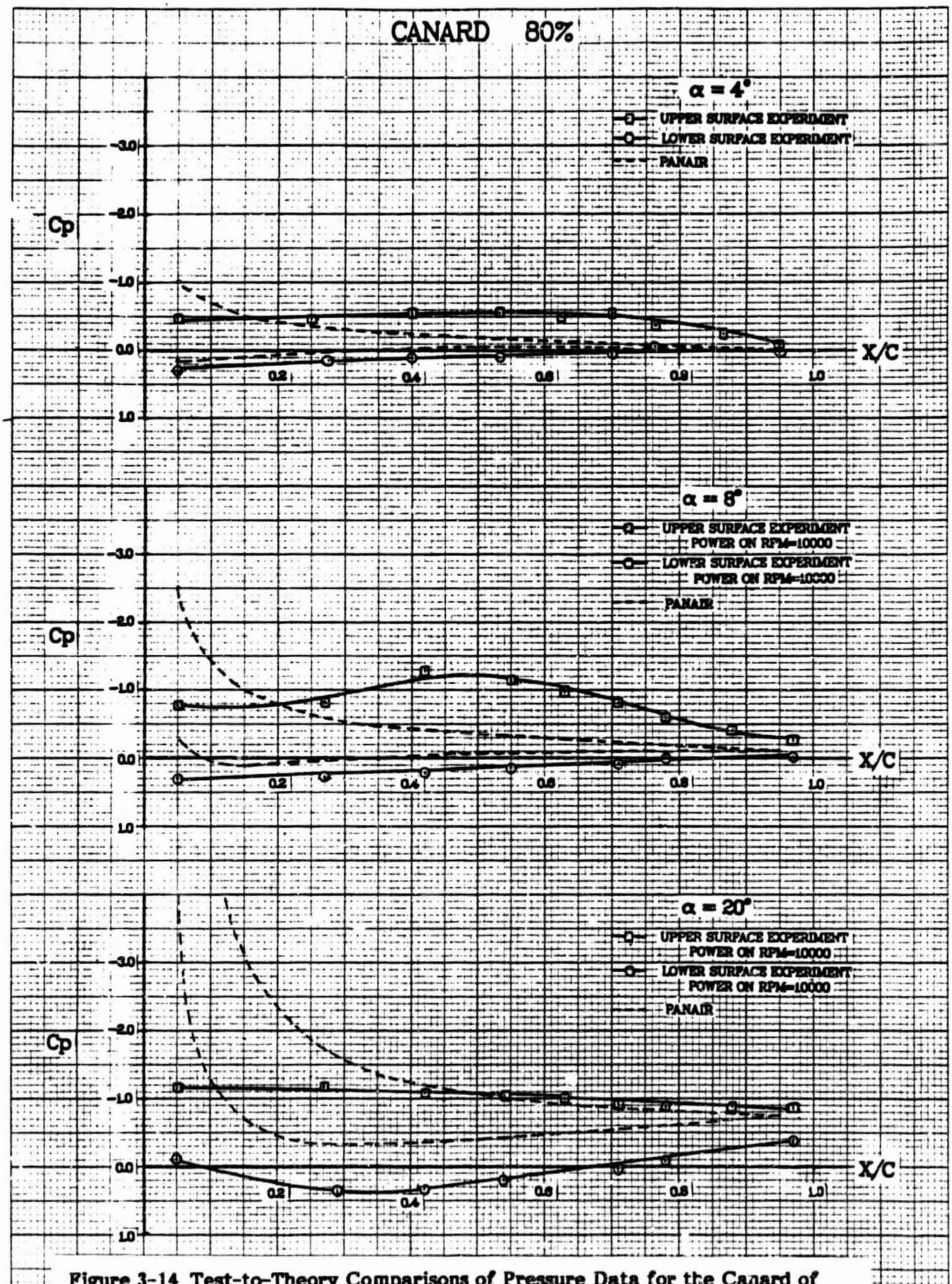


Figure 3-14 Test-to-Theory Comparisons of Pressure Data for the Canard of the Ames/Vought Model, 80% Semispan

ORIGINAL PAGE IS
OF POOR QUALITY

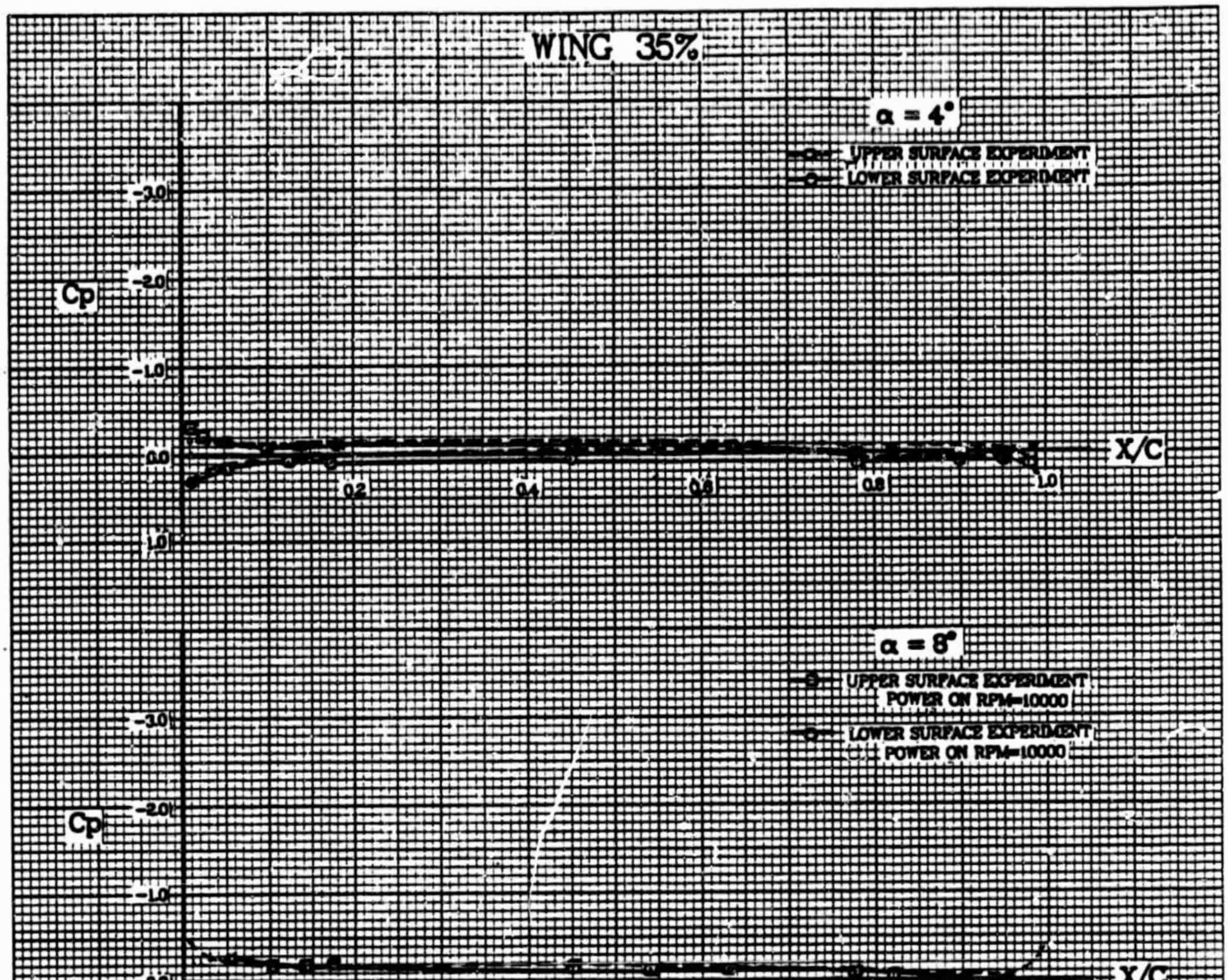


Figure 3-15 Test-to-Theory Comparisons of Pressure Data for the Wing of the Ames/Vought Model, 35% Semispan

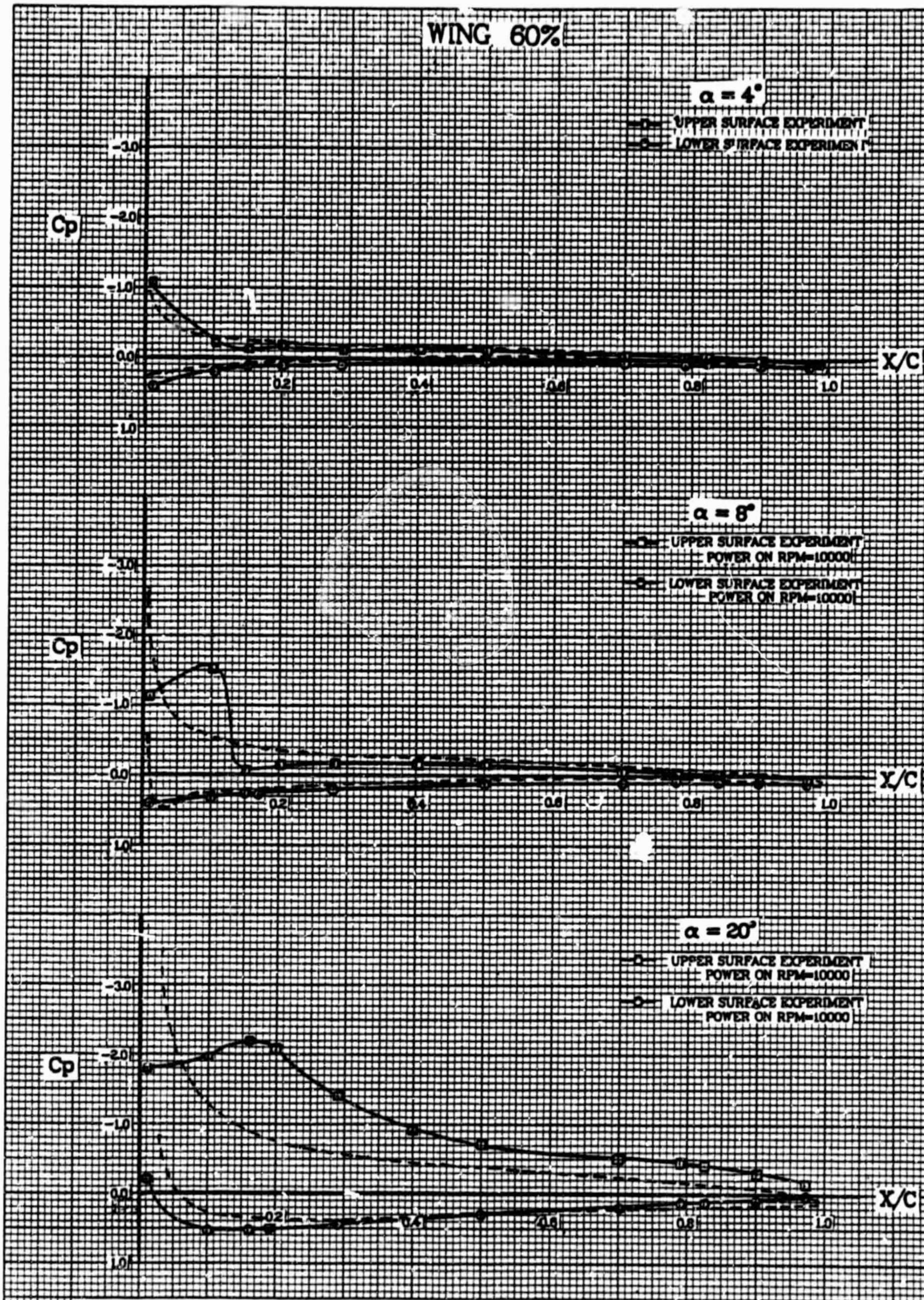


Figure 3-16 Test-to-Theory Comparisons of Pressure Date for the Wing of the Ames/Vought Model, 60% Semispan

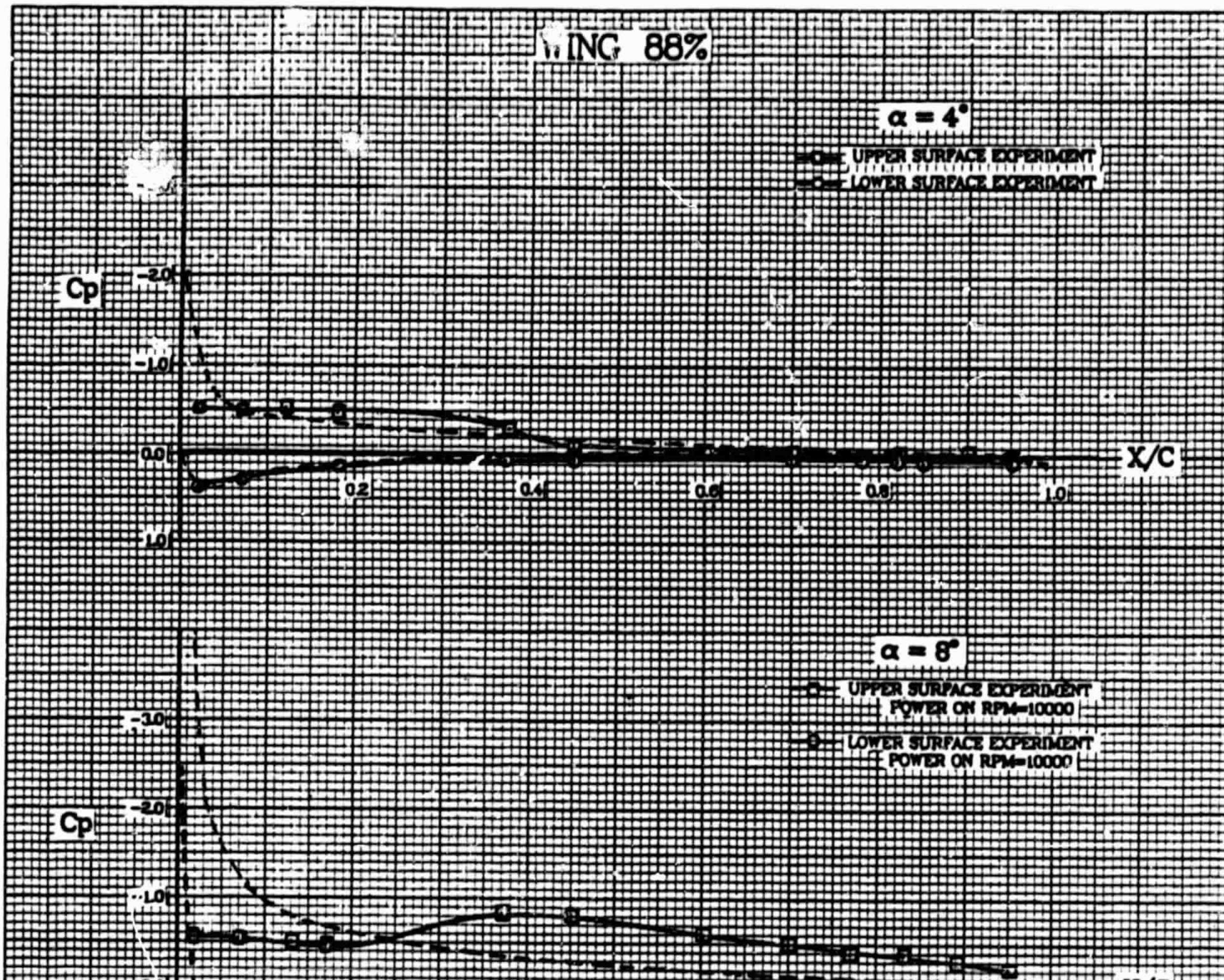


Figure 3-17 Test-to-Theory Comparisons of Pressure Date for the Wing of the Ames/Vought Model, 88% Semispan

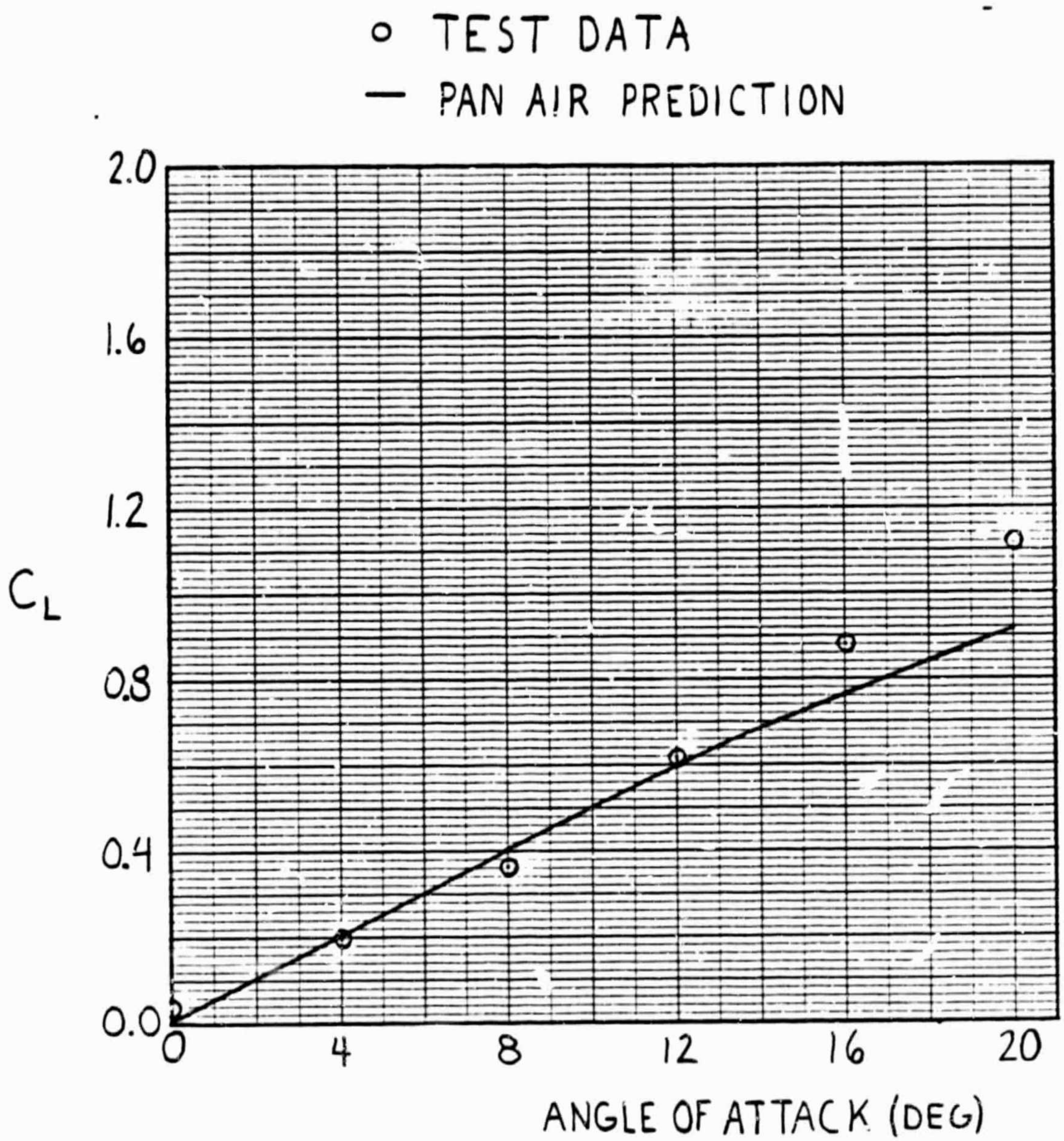


Figure 3-18 **Test-to-Theory Comparison of the Lift Curve for the Ames/Vought Model**

4. PREPROCESSOR MODULE

The Preprocessor Module is intended to be an automated input generation module. This module assists the user in preparing the detailed geometric input required by the PAN AIR program. The Preprocessor was formulated as an interactive graphics program designed to operate on one of the NASA Ames' VAX 11/780 computers. A description of this module including existing methods selected for incorporation into the module is presented in this section. User Instructions, sample input and listing of the code developed during this study are included as Appendix A. The source listing of the existing codes are not included. For these the reader is referred to the references.

The primary function of this module is to generate the geometric panelling arrangement for an arbitrary configuration consistant with the requirements of the PAN AIR code. The generation of a panelling scheme from the geometrical data for an arbitrary three-dimensional configuration is a very tedious, time-consuming task. Errors are easily introduced by the tedious nature of the task, and once introduced are hard to locate. Changes to the panelling arrangement are difficult and therefore, often reluctantly made. The Preprocessor module geometry package coupled with an appropriate plot package solves most of these problems.

4.1 DESCRIPTION OF THE GEOMETRY PACKAGE

The geometry package selected for the Preprocessor Module is based on a panelling code developed at McDonnell Douglas by Halsey and Hess (Reference 28) under contract to NASA Langley. This program was formulated to generate a panelling scheme consistent with the requirements of the three-dimensional lifting version of the Douglas-Neumann Program. The Halsey-Hess geometric procedure was chosen for this application because it was developed specifically for use with a 3-D potential flow program, and it was sufficiently general so that it could be used for the present application with only minor modifications.

The panelling requirements for PAN AIR are similar to those for which the selected geometry package was developed; however, the dimensions set within the program had to be increased to be compatible with PAN AIR. The coding was also altered for

compatibility with the NASA Ames' VAX 11/780 computers. A brief description of the procedure and the options available are presented in the following paragraphs.

The geometry package was designed to simplify the user's job by automating the panel generation. Input to the geometry code consists of a very sparse set of coordinate data to describe each component of a configuration. Often it is possible to describe a component with an order of magnitude fewer points than required for an accurate potential-flow solution. The user is also not required to devote a lot of time and care to develop consistent spacing between the points because the package provides automatic spacing options that augment and redistribute the input points. The package also has the capability to calculate the intersection curves between components and then to repanel the components so that the adjacent paneling distributions are compatible.

Use of the geometry package and understanding what the package does requires that the user be familiar with some terms and conventions used in the development of this code. These terms are illustrated in Figure 4.1 taken from Reference 28. Each component is defined by specifying points on a section curve, then on an adjacent section and so on, until all sections have been defined. Each component thus consists of a set of points which can be connected to form a network of intersecting lines called N-lines and M-lines. Points on a section curve are ordered along the perimeter of the section and are designated as N-lines. On a wing, for example, the N-lines correspond to defining the airfoil shape at a chordwise cut through the wing. The curves connecting corresponding points on the N-lines are designated M-lines. On a wing, the M-lines run spanwise; whereas on a body the M-lines generally run along the length of the body. The area bounded by two adjacent N-lines is termed a strip and the area bounded by two N-lines and two M-lines is termed an element (i.e., a panel). An understanding of N-lines and M-lines is required to comprehend the input terminology of the geometry package.

First, it is necessary to understand generally what the geometry code does before discussing the restrictions and options applicable to the M and N lines. The first operation performed by the geometry package is to panel or redistribute the points defining each component as if each component is completely independent of the others. The paneling of each component is performed in the following two steps:

- o the points on the initial N-lines are augmented in number and redistributed according to the number and the spacing algorithm specified by the user;

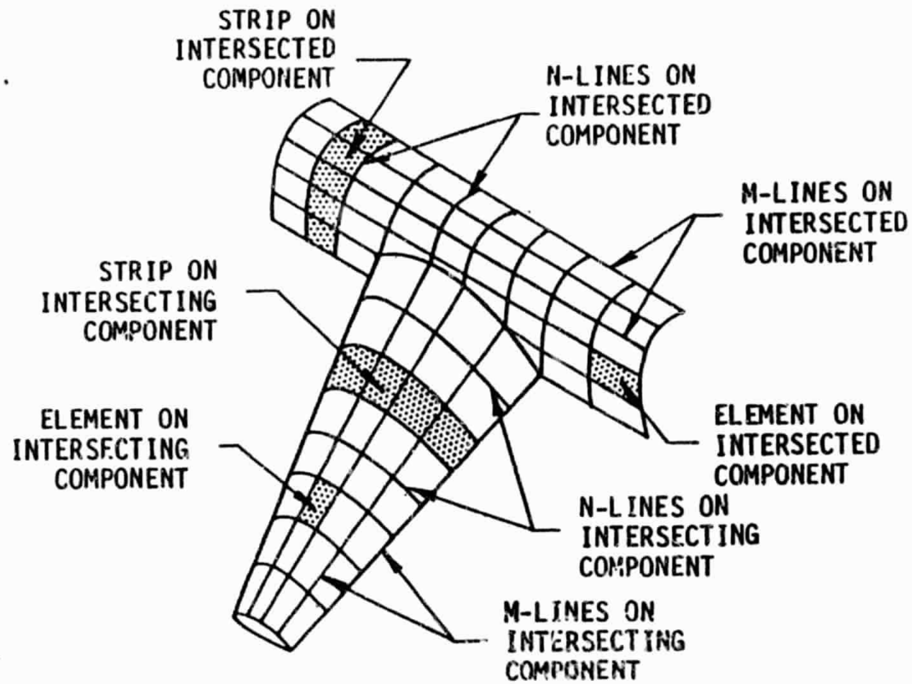


Figure 4.1 Definition of Geometry Package Terms

- the N-lines are augmented in number and redistributed in number according to the user's specifications. The latter is done by augmenting and redistributing points along each M-line.

Cubic polynomials are used for the interpolation required to redistribute the points along either N-lines or M-lines. The arc length along the polygon formed by connecting straight lines between adjacent points is the parameter used as the independent variable of interpolation. Interpolation for a point on a curve is accomplished by three separate interpolations, one for each coordinate. In each interpolation, the first derivative of the dependent variable with respect to the straight-line arc length is first found by taking a weighted average of the angles of the straight-line segments. These are then used, together with the coordinate values, to determine the coefficients of cubic interpolating polynomials.

Both cubic splines and cubic polymonials were investigated for inclusion in the geometry package. The cubic-polynomial method selected is not a true spline method because the second derivatives are discontinuous. Cubic splines were found to cause problems when the defining data was very sparse. Any waviness that might exist in the defining data can get out of control when using cubic splines to interpolate a very sparse set of data.

Several options for the spacing of points on N-lines (i.e., in the chordwise direction on wings and usually the circumferential direction on fuselages) are available for the user to select. These options are as follows:

- Input distribution, unaltered
- Input distribution, augmented in number
- Constant increments in arc length
- Cosine spacing
- Curvature dependent distribution
- User specified distribution

No interpolation is done when the unaltered input distribution option is selected. This option may be used when the initial defining distribution contains a sufficient number of properly spaced points. The user must insure that the points are properly spaced such that when M-lines are formed, the distribution of points on adjacent N-lines are fairly similar resulting in smooth curves.

The input distribution, augmented in number option results in a distribution which is similar to the initial, defining distribution, but contains a different number of points. The user must input properly distributed points, but he need not input an extremely large number of points. This option is useful for controlling the desired distribution without having to load a large number of points and without having to calculate what the numerical values of the arc lengths are.

The constant increments in arc length option results in points distributed uniformly around the perimeter of the N-line. This option is useful for smooth bodies that do not have large variations in curvature. Use of this option for a wing component is not recommended because it would result in too few points near the leading and trailing edges to adequately represent the wing for potential flow calculations.

The cosine spacing option results in a very useful distribution for wing components. This option results in a very fine spacing near the leading and trailing edges and coarse spacing in regions away from these edges.

A very useful option is the curvature-dependent option where the point spacing is a function of the local surface curvature. The results of potential flow methods are improved by using fine spacing in regions of high curvature. In this option, the spacing is linearly dependent on the absolute value of the curvature. This selection is useful for both wing and body components because some control over the variation of the spacing is provided. When this option is used for a lifting component, artificial curvature is added in the usually flat trailing-edge regions resulting in more points bunched on the trailing-edge as required by potential flow methods.

If none of the preceding options are adequate, the user can specify a distribution. This distribution is specified in the form of normalized arc length ($0.0 <$ arc length < 1.0).

There are four spacing options available for N-lines (spanwise spacing for wing and lengthwise spacing for fuselage). The redistribution of points on the initial N-lines results in a situation where each N-line has the same number of points distributed in a similar manner. A set of M-lines is generated by connecting the corresponding points on all N-lines. The M-lines should be smooth and have fairly small curvature since the distribution of points on each N-line is the same. The process of augmenting and redistributing the N-lines is accomplished by augmenting and redistributing points along these M-lines. It is

required that each M-line have a total length greater than zero. The four options available for the spacing of N-lines are as follows:

- o Input distribution, unaltered
- o Input distribution, augmented in number
- o Constant increments
- o User-specified distribution

These spacing distributions are completed analogous to the preceeding options for distributions of points on N-lines.

There are two modes of operation of the geometry package during the distribution of the N-lines. These two modes are the planar-section mode and the arc-length mode. The planar-section mode, which is the most commonly used mode, redistributes the points in such a way that all N-lines are parallel except perhaps the first and last ones on a component. In the arc-length mode, the points along M-lines are distributed in such a manner to allow the N-lines to form smooth curves around a nonplanar edge of the component. This is useful for a case where one edge of a component may be so highly nonplanar that a strip bounded by that edge and an adjacent planar N-line would vary drastically in width. The arc-length mode would allow the adjacent N-lines to form smooth curves nearly parallel to the nonplanar edge.

After paneling the isolated components, the next major operation performed by the geometry package is the calculation of the intersection between components. The intersection curves between components are calculated using a hybrid curve-fit/surface-fit approach. A set of intersection points is obtained by calculating where the M-lines of one component (designated the intersecting component) pierce the surface of the other component (designated the intersected component). The method calculates the intersection point of a curve and a surface. After the intersection is determined, the intersecting components are repaneled so that adjacent panels on either side of the intersection curve line up in a manner which is satisfactory for potential flow calculations.

There are some limitations and restrictions that apply to the calculation of intersection curves between components. This method is satisfactory for well defined wing-fuselage intersections, wing-pylon intersections, etc.; however, it is not designed to

handle complex cases where the intersection curve is discontinuous or is not well defined due to fillets or smooth transitions from one component to another. Guidelines to follow when calculating the intersection between components are summarized (from Reference 28) as follows:

- A distinction can be made between intersecting and intersected components.

- A component can intersect only one other component and can be intersected by only one other component. If a body intersects or is intersected by several bodies, it must be divided up into several components. A single component can intersect one component and be intersected by another component, however.

- The M-lines on the intersecting component pierce the surface of the intersected component at a sharp angle. Surfaces are not tangent where they are meet.

- The intersecting component has at least one N-line which lies entirely in the interior of the intersected component.

- No M-line on the intersecting component intersects the intersected component more than once.

The final function of the geometry package is to repanel all components that intersect other components. Some important features of the final repaneling of components are:

- That portion of any component which falls inside another adjacent component is eliminated or at the user's discretion designated as extra strips or ignored.

- The entire component is repaneled to insure a smooth distribution of the paneling over the entire component.

- The user may specify a repaneling option with exact panel matching between the components or panel matching at every second point on the intersection curve.

- All intersecting components (typically the wing in a wing-body intersection problem) are repaneled the same way whether they are lifting or nonlifting components. There is no redistribution of points along N-lines, only a redistribution of N-lines. A new N-line is added along the intersection curve in every case. The remaining N-lines on the exterior of the component are respaced to avoid irregularities in the widths of the strips.

- The repaneling of a nonlifting intersected component in an intersection problem was designed primarily for fuselages. Three options are provided for repaneling of nonlifting intersected components. The first is no repaneling at all. The second and third options are designed to produce panel distributions that depend upon the paneling scheme of the intersected component. With each of these options, the body is divided into four components. These components are defined as that portion of the body forward of the leading-edge point of the intersection curve, that portion of the body aft of the trailing-edge point on the intersection curve, that portion of the body above the intersection curve, and that portion of the body below the intersection curve. The last two options differ only in the distribution of panels in the regions above and below the intersection curve. Matching with every panel or every other panel on the wing can be selected by the user.

- For the repaneling of lifting intersected components, the user also has three options. The first is to do no repaneling. The second defines a region of ignored elements around the intersection curve but results in gaps in the region of the intersection curve. The third is similar to the second except that the gaps are filled with nonlifting panels.

Knowledge of the preceeding terminology and restrictions is essential for understanding the geometry package utilization presented in Appendix A. The primary output of the geometry package is a VAX file containing the desired paneling definition of all of the aircraft components. The format of the coordinates is acceptable for input to PAN AIR when the appropriate JCL and boundary condition definitions have been added. It may be desired to divide some of the components into several different networks for the PAN AIR problem. Currently, this must be handled by the user using the VAX editor mode.

4.2. GRAPHICS

The coupling of the geometry package with a graphics package is a very important capability. This allows for easy verifying and editing of the geometry files or PAN AIR input files. With the graphics package, the paneling scheme may be rotated to any desired viewing orientation and scaled to any desired viewing size. The entire configuration, any individual component (network, if plotting PAN AIR input file), or any combination of components may be viewed. Another important feature incorporated into the graphics package is a hidden line removal option reported in Reference 29. This feature adds a lot of clarity to any particular view thus allowing easier identification of any problem areas needing editing. This feature is illustrated in Figure 4.2 showing a detailed panel arrangement generated for PAN AIR. User instructions for the graphics package and the code developed for this package are presented in Appendix A.

4.3 PANELLING OF VOUGHT VATOL FIGHTER MODEL

The panelling scheme used for the analysis of the Vought VATOL Model using the PAN AIR code discussed in Section 3.2 was prepared by exercising the preprocessor module. The panelling definition that was available as a starting point was unsatisfactory for input to the PAN AIR code. To accomplish this repanelling by hand calculation and manipulation would have been a time consuming and error prone task. Using the preprocessor geometry package linked with a graphics package significantly reduced the tedious tasks to be done by hand and provided a quick and easy way to verify the final panelling scheme. With these tools, several panelling arrangements were investigated before choosing the one deemed most desirable.

The starting and the final refined panelling scheme for the Vought model are shown in Figure 4.3. In addition to several minor coordinate errors that were easily detected with the graphics package, the initial panelling contained several areas that would have had led to problems with the PAN AIR code. The chordwise cuts on the wing and canard were not aligned and the panels along the fuselage and nacelle did not always match. In addition at the abutment of the different components there were mismatches between adjoining panels. It was desirable to have at least two panels between the canard wake and wing surface; the initial panelling had only one. These problems would have resulted in PAN AIR failures and/or would have resulted in questionable PAN AIR predictions. The

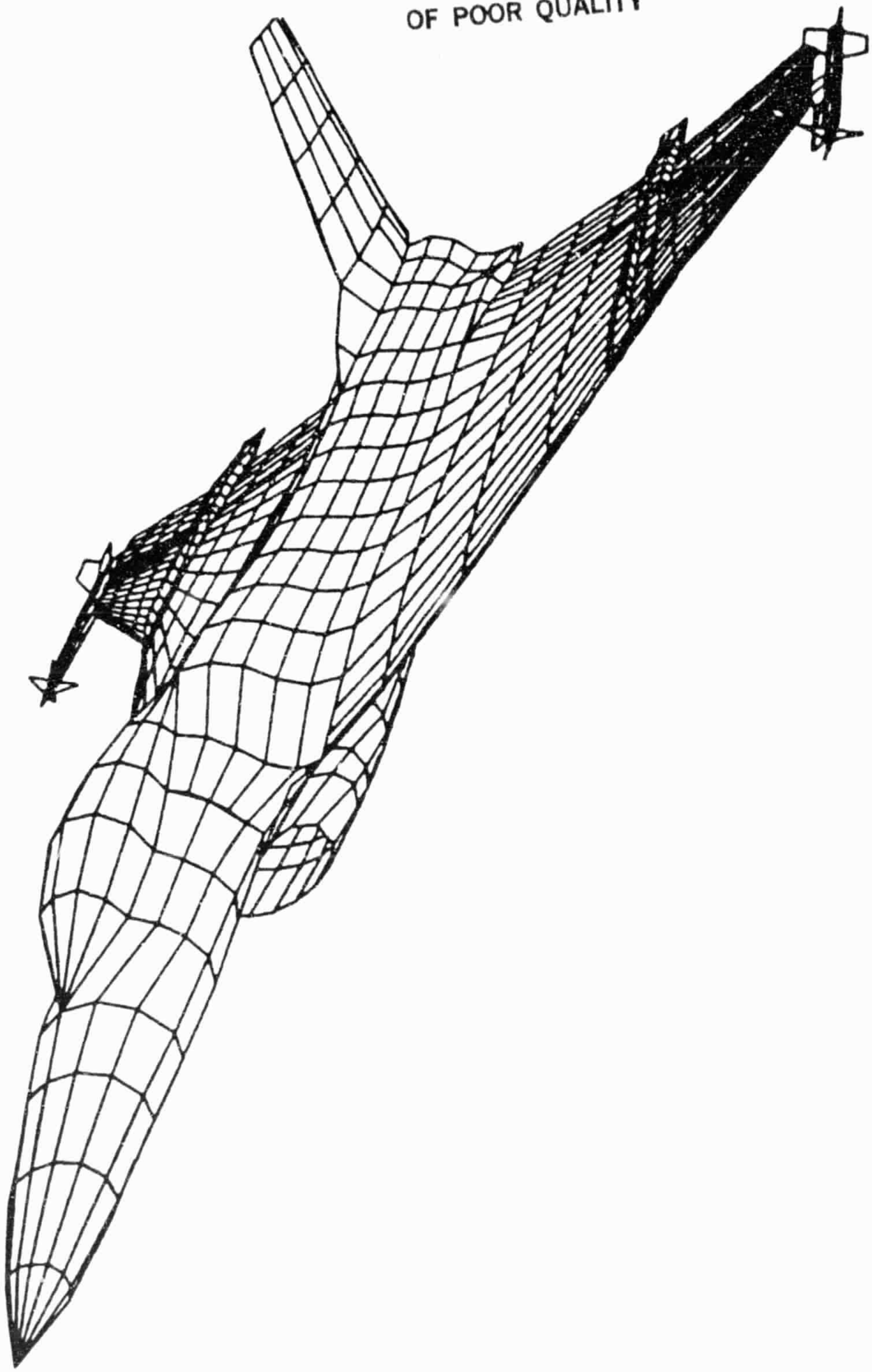


Figure 4.2 Example of Graphics Package

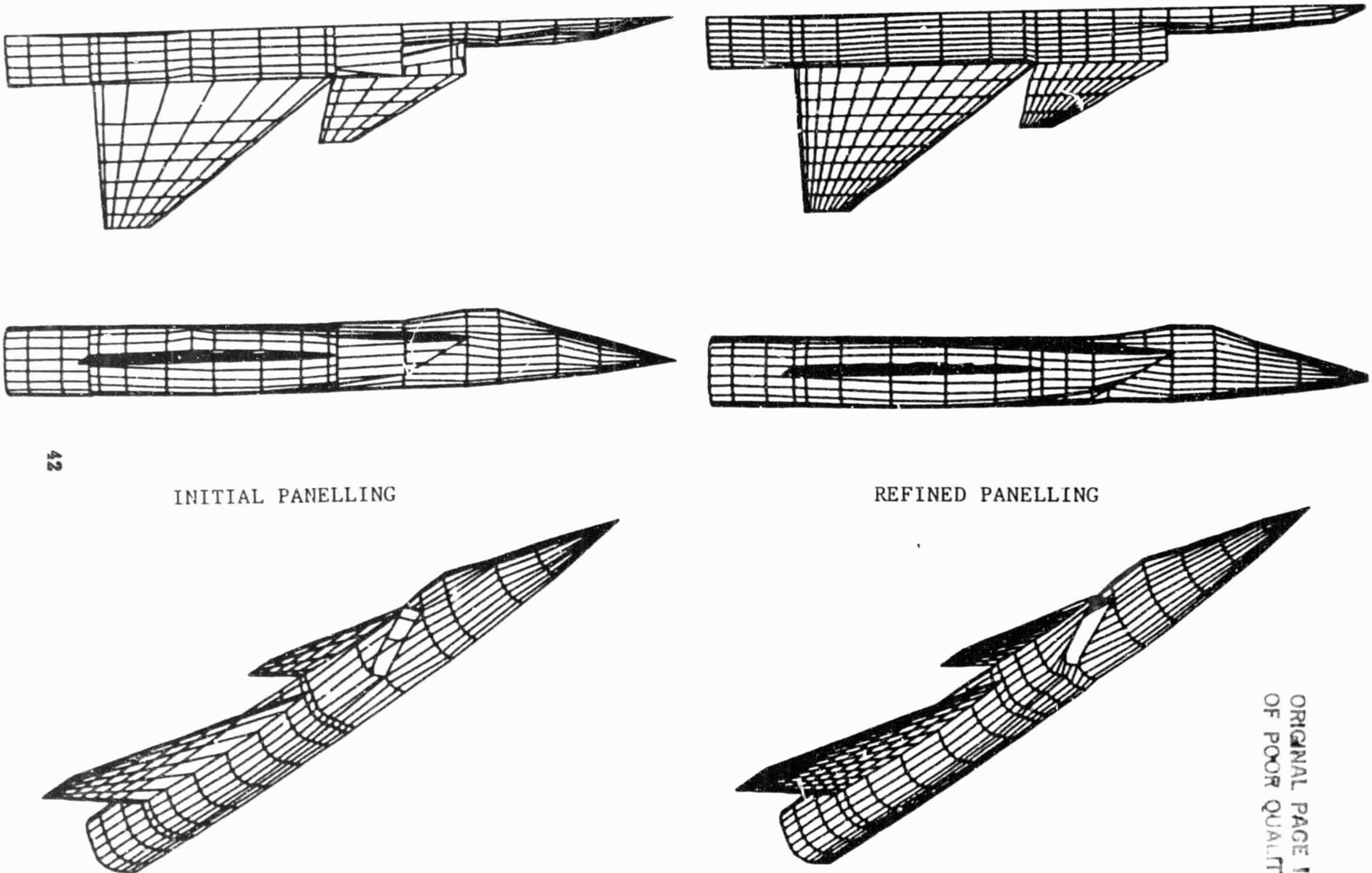


Figure 4-3 Initial and Final Panelling Schemes for Vought VATOL Model

spacing options in the geometry package allowed the user to easily specify the spacing on each component thus eliminating the problems. The intersection option in the geometry package was also used to ensure matching between the fuselage and wing.

The final panelling scheme for the Vought VATOL model as input to the PAN AIR code is shown in Figure 4.4. The wake networks were removed for clarity. The PAN AIR predictions for this arrangement were presented in Section 3.2.

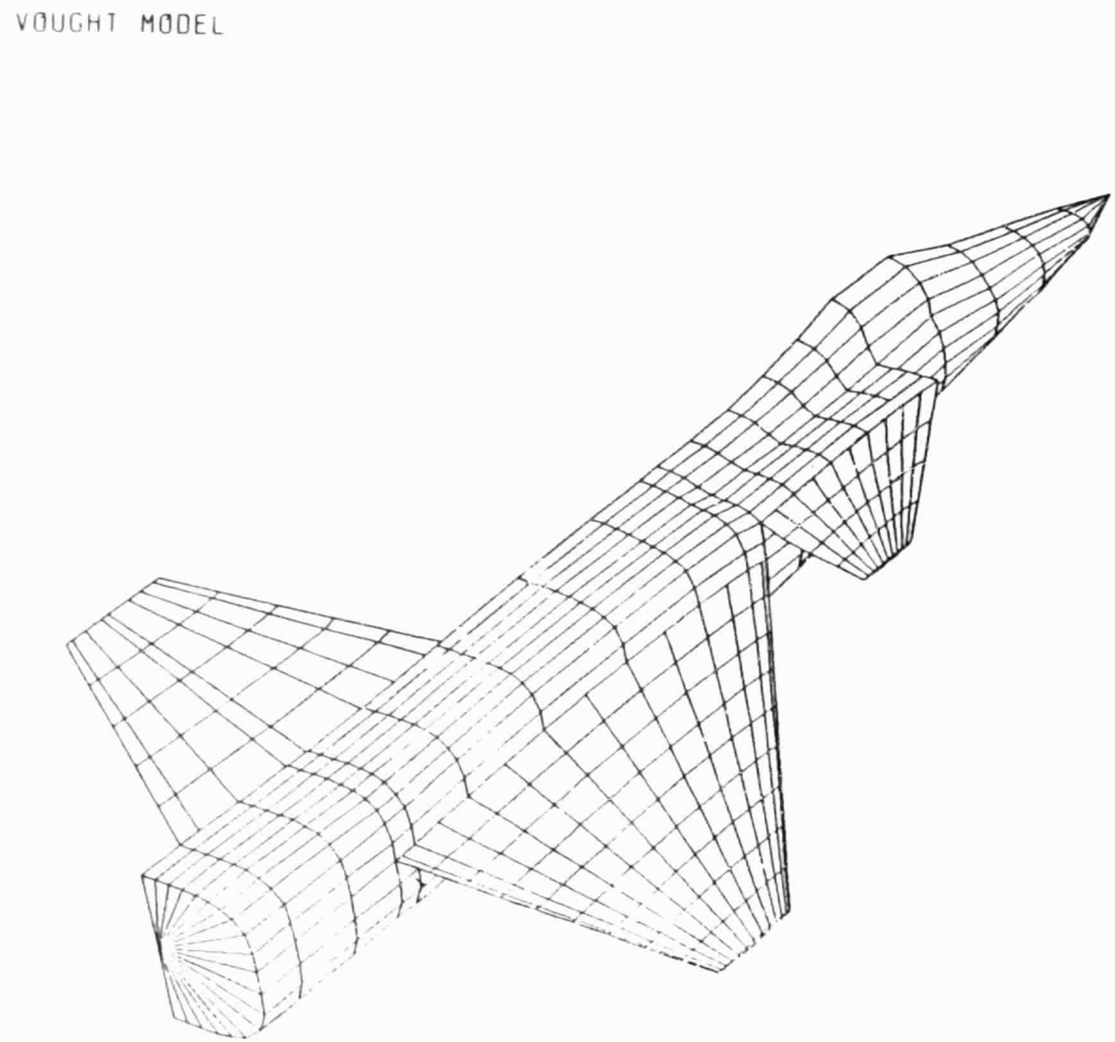


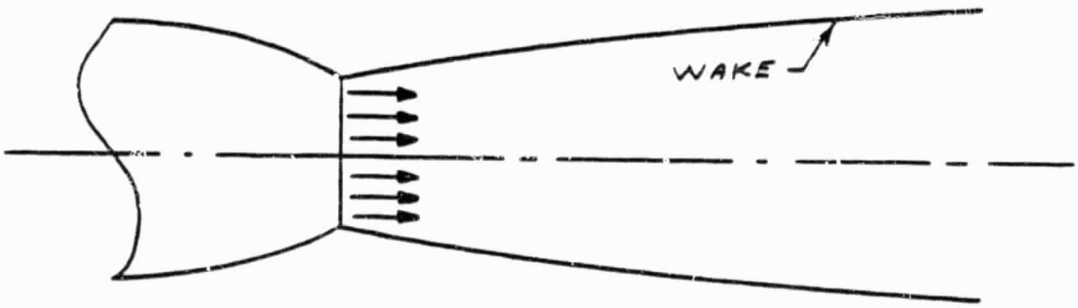
Figure 4-4 PAN AIR Panelling Scheme for Vought VATOL Model

5. ENGINE SIMULATION MODULE

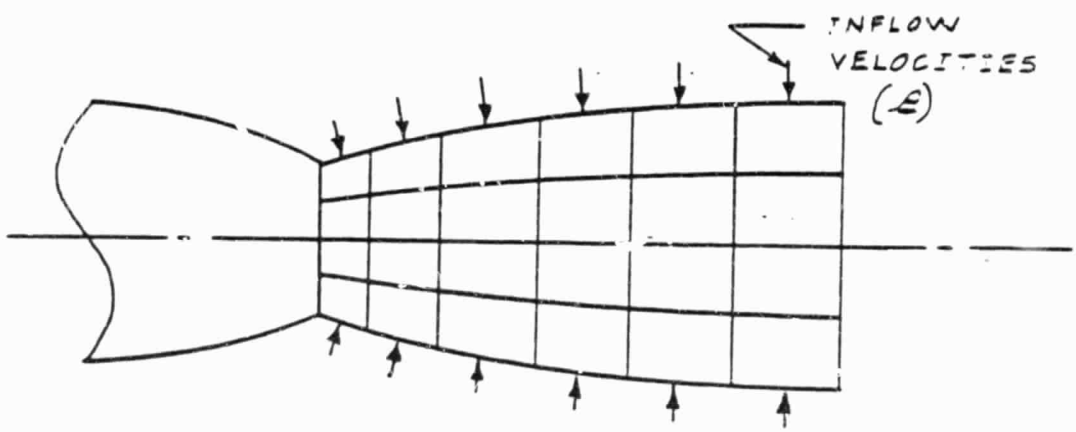
Modern aircraft that are designed for STOL operation often supplement aerodynamic lift with direct thrust from the propulsion system or from other devices dedicated to the production of direct lift, such as lift jets. In either case, there is an injection of high velocity air and/or exhaust gases into the freestream flow. This flow condition is also experienced by VTOL aircraft in transition between hovering and conventional flight. At the low speeds associated with this type of flight, the jet velocity to freestream velocity ratio is relatively high, and hence, the interference phenomena between the two flowfields can have a significant influence on the aerodynamic forces on the surface of the aircraft.

The Engine Simulation Module is designed to include techniques that will simulate flow perturbations caused by the propulsion system in an aerodynamic potential flow analysis code. In a previous study (Reference 20) two basic ideas for modeling power effects in the PAN AIR code were evaluated. The first approach, which is illustrated in Figure 5-1a required specification of the velocity or mass flux boundary conditions on the nozzle exit plane. It was hoped that the aerodynamic solution performed by PAN AIR, with high velocities specified at the nozzle exit, would simulate a plume and cause the desired flux perturbations in the external flowfield. Wake networks in the approximate shape of a plume surrounding the exit would be utilized to sustain the high velocities downstream of the exit plane. Several variations of boundary conditions were investigated to implement this approach. It was found that some of the combinations of boundary conditions were not properly handled by the PAN AIR code. In other cases that were apparently handled correctly, mathematically speaking, the velocity downstream of the exit plane decayed too rapidly. In fact, computations indicated that the exhaust velocity decreased to almost freestream velocity within one nozzle exit diameter. Thus, it was concluded that a plume could not be satisfactorily simulated by specifying the velocity on the exit network.

The second approach addressed in Reference 20 was to represent the plume with permeable networks to represent the boundary. Flow entrainment was simulated by specifying a velocity normal to the panels of the plume. This approach is illustrated in Figure 5-1(b). Plume modeling methods must be used external to the PAN AIR code to derive the shape and boundary conditions that represent the plume. The PAN AIR code



(a) VELOCITY SPECIFICATION APPROACH



(b) PERMEABLE BODY APPROACH

Figure 5-1 Techniques for Modeling Exhaust Jets

appeared to correctly analyze the configurations with plumes modeled in this manner. Therefore, the study reported in Reference 20 recommended this approach to modeling jet exhaust flow in the PAN AIR code.

In the current study, a method was selected for use in representing plumes with permeable networks. The conditions addressed are limited to supersonic exhaust jets in a subsonic freestream flow. The exhaust may be aligned with or slightly inclined to the freestream flow. Although the primary objective has been to develop methodology for low-speed applications, this method may eventually be used throughout the subsonic Mach range. This section of the report discusses the plume modeling method that was selected and the approach that was taken to incorporate it into the Engine Simulation Module. User instructions and a sample problem are presented in Appendix B.

5.1 SELECTION OF THE PLUME MODELING METHOD

A literature survey was conducted to find existing plume modeling codes that could be adapted for use in the Engine Simulation Module. Several methods were found that can be used for predicting the effects of exhaust plumes on the configuration afterbody. Putnam and Mace (Reference 3.0) summarize these methods and present a discussion of two approaches to the solution of nozzle exhaust flow. The first method is based on a solution of the Navier-Stokes equations and the second is based on a patched inviscid/viscous interaction approach.

The computation of jet exhaust flowfields based on the solution of Navier-Stokes equations is now possible due to the availability of faster computers. These solution techniques are now being applied to both two-dimensional and three-dimensional problems. However, the application of these methods is not straight forward and requires familarity with a variety of techniques to obtain a valid solution. Even with these extended efforts, the large computer time required for a Navier-Stokes solution preclude its use for the analysis of complete configurations including jet effects.

Patched inviscid/viscous solutions to plume analysis, which have been shown to provide good predictions of the plume characteristics as well as the nozzle surface pressure distributions, are easier to use and require significantly less computer time than the Navier-Stokes solutions. Their approach to solving the nozzle flow problem is to

divide the plume and external flow into several distinct regions, each of which can be analyzed independently. The various regions are analyzed in an iterative manner until convergence is obtained.

Two of the foremost plume modeling codes that use Navier-Stokes and patched inviscid/viscous solutions are the Cline code and the Wilmoth code, respectively. These methods are presented in References 31 and 32. An isolated nacelle model with a specified set of exit conditions was analyzed with each of these codes. A comparison of the computed flow properties are compared at the nozzle exit (X=6.7) and at two downstream stations in Figure 5-2. These properties match reasonably well, except for the radial velocity at the most downstream station (X=15.8). The Navier-Stokes solution had been computed for 2,000 iterations for the data shown here, and it is noted that the solution for the radial velocity had not completely converged. Based on the plume predictions discussed above, and computer resource requirements, the Wilmoth code was selected for use in the Engine Simulation Module.

The Wilmoth code, which is also known as RAXJET, performs a two-dimensional viscid analysis of a region that includes both an external flow and an exhaust flow. Several assumptions were made (to be discussed in Section 5.3) in order to adapt it to the Engine Simulation Module. One important consideration in the selection of this code is that it is currently being developed into a full three-dimensional procedure through funding by the NASA Langley Research Center. RAXJET's capability to predict the effects of viscosity in the external flowfield is not currently being used.

5.2 DESCRIPTION OF THE RAXJET PROGRAM

The RAXJET code is comprised of methods that evolved over a number of years through the efforts of several researchers. These methods are the South-Jameson relaxation external flow analysis procedure, the Reshotko-Tucker boundary-layer solution, the Presz separation model, the Dash-Pergament mixing model, and the Dash-Thorpe inviscid plume model. These computational techniques are coupled iteratively into a single computer program called RAXJET. This program predicts subsonic and transonic axisymmetric flow over nozzle afterbodies with supersonic jet exhausts and accounts for boundary-layer displacement, separation, jet entrainment, and inviscid jet plume

K•E 10 X 10 TO 1/2 INCH 7 X 10 INCHES KEUFFEL & ESSER CO MADE IN U.S.A.

46 1320

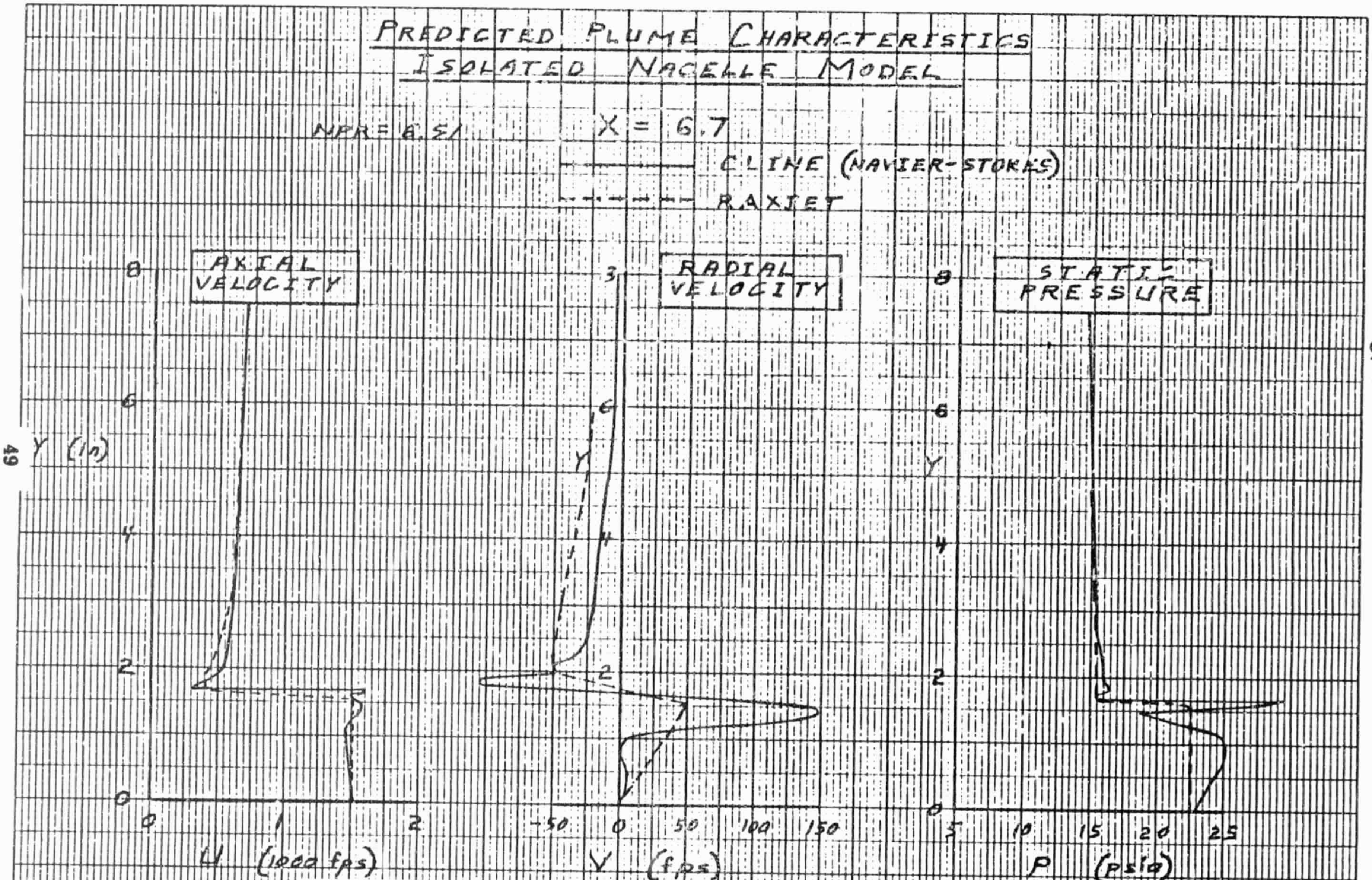


Figure 5-2 Comparison of Plume Characteristics Computed by a Navier/Stokes Solution and a Patched Inviscid/Viscous Interaction Solution

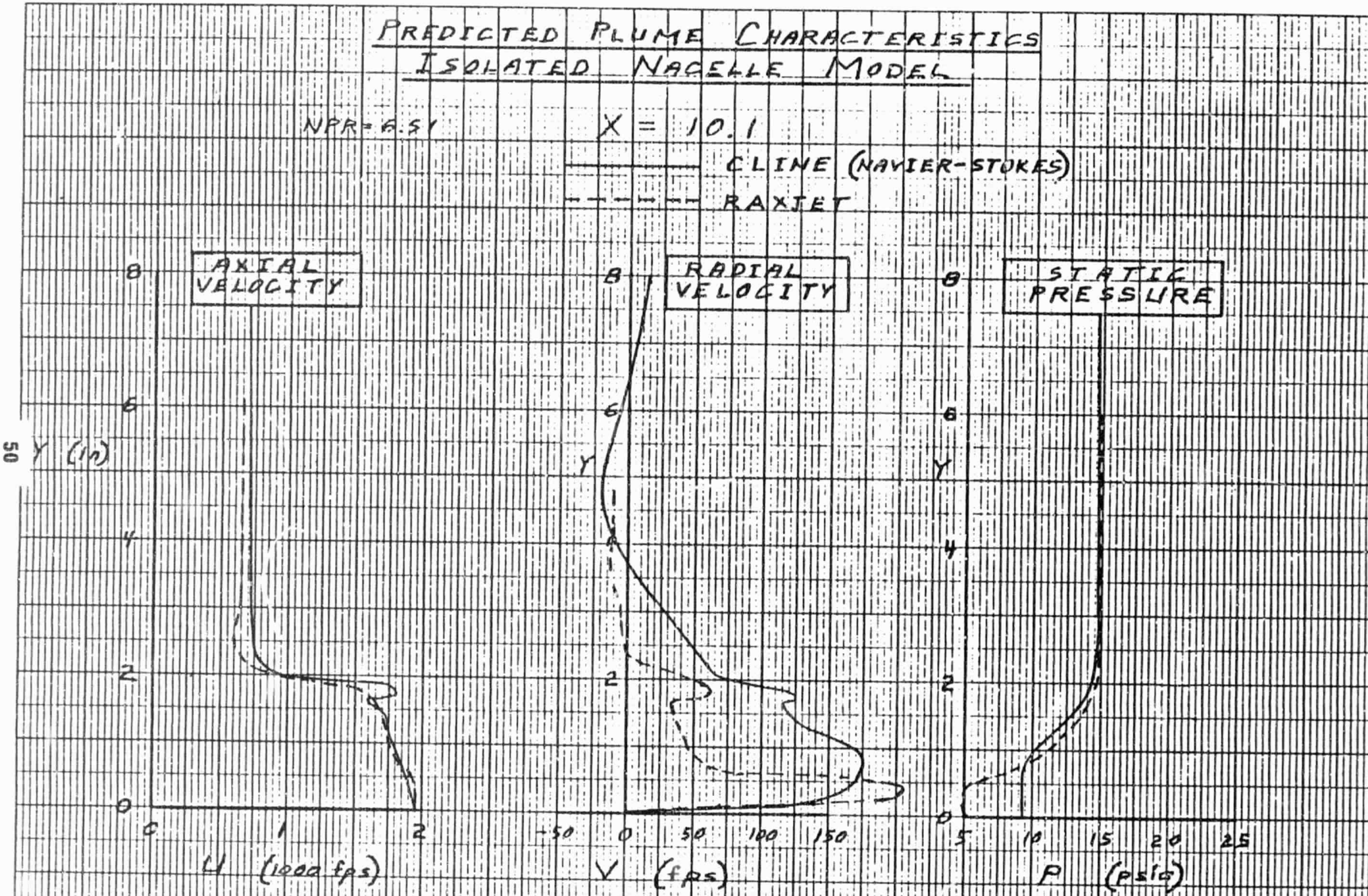


Figure 5 - 2 (Continued)

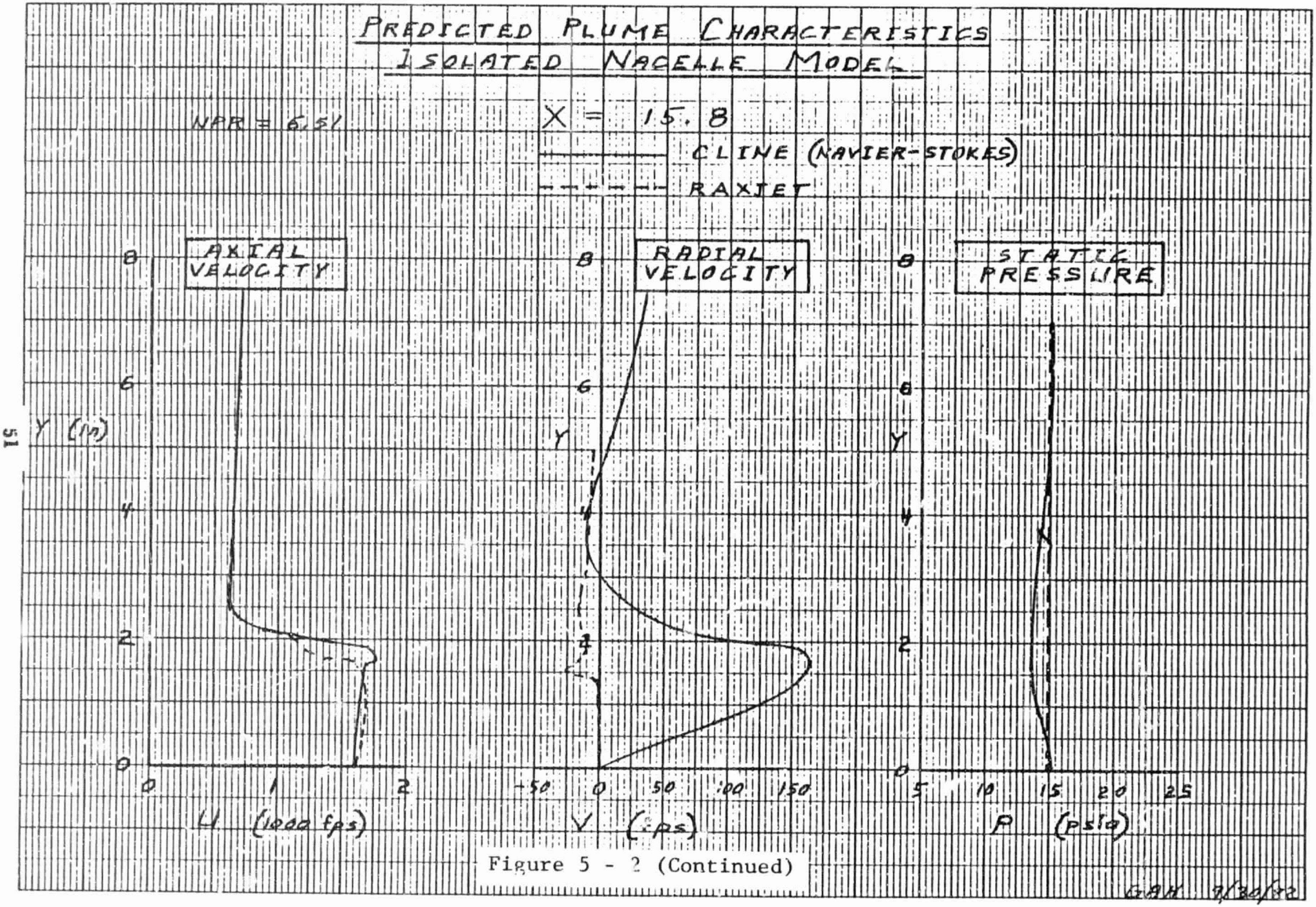


Figure 5 - 2 (Continued)

blockage. The various computational domains where different solution algorithms are employed in RAXJET are illustrated in Figure 5.3 and are summarized below.

Inviscid External Flow Solution - The relaxation procedure of South and Jameson (Reference 33) is used for solving the exact nonlinear potential flow equation in nonconservative form. A computer code developed by Keller and South, commonly known as RAXBOD, (Reference 34), is used to implement this technique.

Inviscid Jet Exhaust Solution - RAXJET uses the computer program SCIPAC (Reference 35) to solve the flowfield in the inviscid jet region through a solution of the conservative finite-difference form of the inviscid flow equations. The nozzle exit flow is assumed to be supersonic, and the calculation is initiated at the exit with the specified exhaust properties. The Mach disk location is determined by the iterative procedure of Abbett (Reference 36).

Boundary-Layer Solution - The properties in the attached boundary-layer region are solved by a modified version of the Reshotko-Tucker integral method for turbulent flow (Reference 37). The integral solution is obtained by conventional boundary-layer marching procedures to yield the displacement thickness distribution over the body using a computer program described in References 38 and 39.

Separated Flow Solution - The Presz Method (Reference 40) is used to predict flow separation and the discriminating streamline shape. The location of separation on the afterbody is calculated by a control volume technique. The discriminating streamline near the separation point on the boattail is calculated by a control volume analysis that solves integral forms of the continuity and streamwise momentum equations to account for streamwise pressure gradients and surface skin friction. The discriminating streamline in the jet wake region is calculated by an integral method that accounts for entrainment effects based on a turbulent mixing analysis (Reference 41).

Mixing-Layer Solution - The mixing layer is the region between the core of the exhaust plume and the external flow. A displacement-thickness distribution arising from entrainment into this layer is calculated by the overlaid mixing model of Reference 42 and the BOATAC computer program (Reference 35). This code solves the parabolic mixing equations by a finite-difference marching procedure.

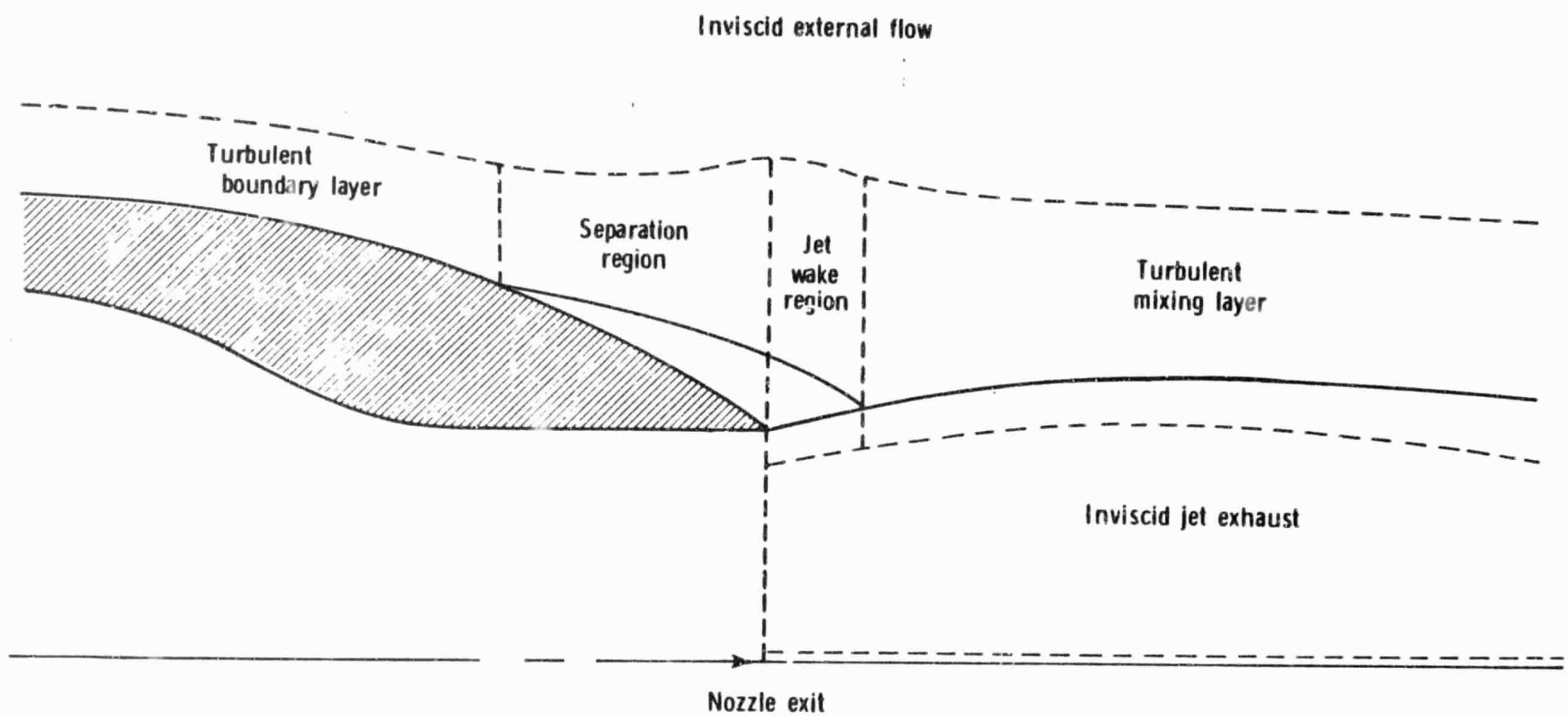


Figure 5-3 Regions of the Patched Viscous/Inviscid Interaction Model

Turbulence is modeled by either a mixing-length or a two-equation transport model. A laminar viscosity option is included. Also calculated are an effective plume boundary that accounts for mass entrainment into the mixing layer and an inviscid plume boundary. The difference between the effective and inviscid plume boundaries define a displacement thickness related only to entrainment.

The various codes discussed above are coupled together in an iterative procedure. The component solutions are performed in the following order.

1. Calculate inviscid external flow field
2. Calculate the inviscid jet exhaust boundary and flow field.
3. Calculate the boundary-layer displacement thickness.
4. Calculate the separation location.
5. Calculate the shape of the discriminating streamline.
6. Calculate the mixing-layer displacement thickness.
7. Calculate a new effective body and plume shape.

These steps are repeated in a single loop in the RAXJET code for a specified number of steps. Convergence is monitored manually by checking surface pressure predictions and/or boundary layer displacement thickness. A thorough discussion of the RAXJET code is presented by Wilmoth in Reference 32.

5.3 MODIFICATIONS TO THE RAXJET PROGRAM

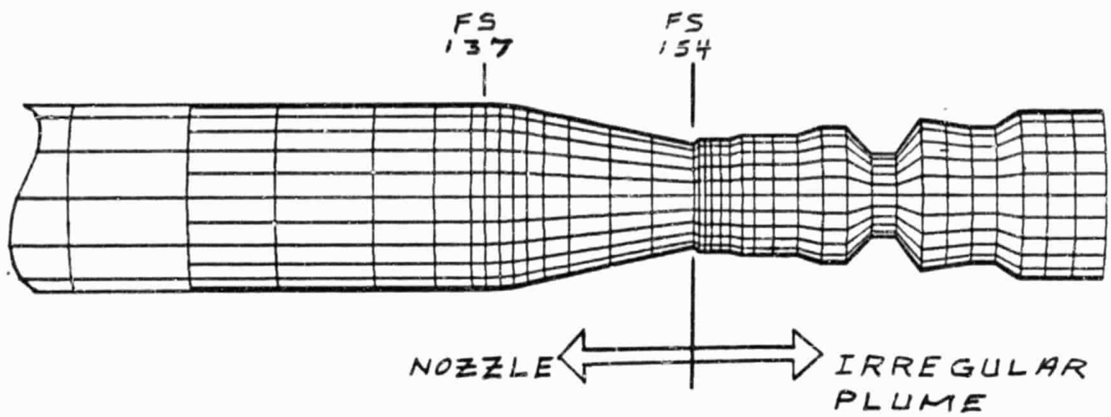
Although the inviscid plume characteristics required for this study could be obtained from selected subroutines in the RAXJET code, the entire code was implemented in the Engine Simulation Module. This not only maintained the integrity of the code but also provided an easily accessible source to a number of parameters that are not currently being used. Two additional capabilities that may be added to the Engine Simulation Module are to (1) account for the influence of the fuselage/boattail boundary layer on the plume characteristics and (2) iteratively solve for the plume shape, accounting for the mutual interference effects between the plume and configuration boundaries. Maintaining the entire RAXJET code provides for this type of expansion without major revision to this portion of the module.

Several modifications were made to the RAXJET code in order to interface it with the Engine Simulation Module. For example, the input subroutine of the RAXJET code and the Engine Simulation Module were merged to avoid repetitious user inputs and several input parameters required by RAXJET were preset within the code to specified values that are consistent with low-speed flight. Also, the nozzle boattail geometry was predefined within the input subroutine. This is permissible since the inviscid plume characteristics are independent of the nozzle boattail geometry, and because the plume boundary is non-dimensionalized in the RAXJET output routines before subsequent use by the Engine Simulation Module.

The primary enhancement made to the RAXJET code was the development of a means of computing plume characteristics that would be meaningful in a PAN AIR analysis. This problem was made even more difficult since there is no unique definition of a plume boundary. Several combinations of plume boundary and inflow velocity were found to be equally effective. An example is taken from Reference 20 of an analysis of a plume shape computed with a Navier-Stokes solution that was defined as the focus of points defining the radial distance (y) at which the rate of change of the axial velocity (dVx/dy) was maximized. This plume shape was very irregular as shown in Figure 5.4. The same reference also shows the results of an analysis made with a smoothed version of this plume shape (Figure 5.5). Comparison between these results show that either shape, although substantially different, can be successfully used in a PAN AIR analysis, provided the corresponding inflow velocities are applied in each case. This is as it should be, since the magnitude of the specified inflow velocity governs the change in the slope at the boundary. As long as the geometrical slope and the change in the slope due to inflow velocity add to the same value, the results computed by the PAN AIR code will be nearly identical.

Since the value used for the boundary of a plume can be somewhat arbitrary, the inviscid plume boundary was selected because it could be easily obtained from the RAXJET code. The radii for this boundary were already computed within the RAXJET code; therefore, it was possible to obtain the inviscid plume boundary by adding a common block within the code to transport the values to an output subroutine.

Entrainment of the external flow into the region of the plume is represented by an inflow velocity that is normal to the inviscid boundary. This value is computed along the



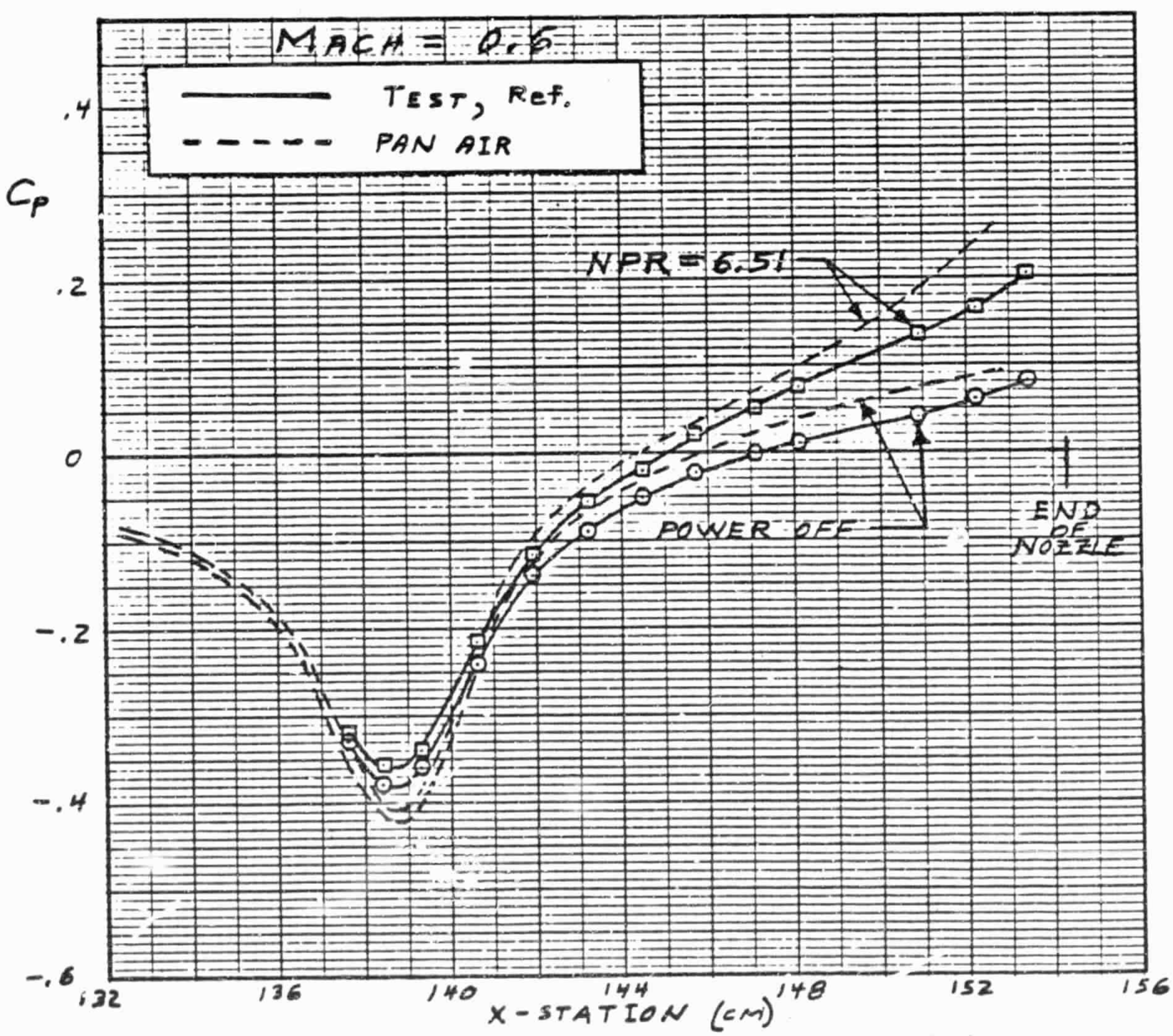
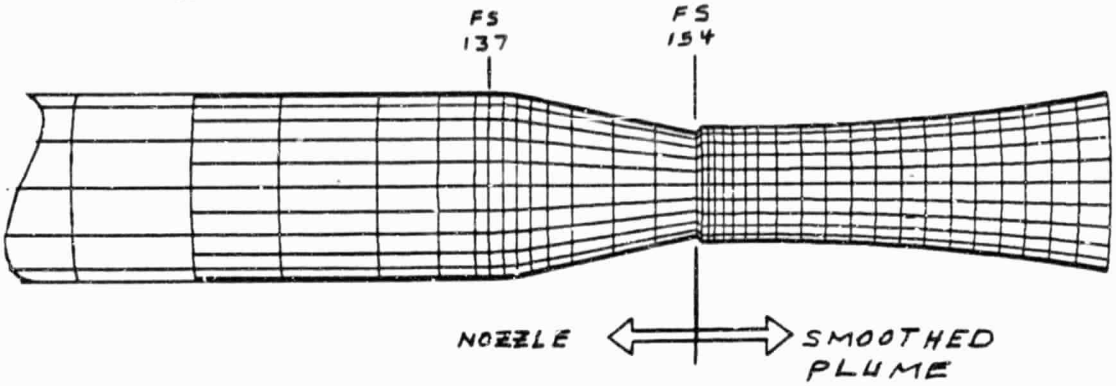


Figure 5-4 A Navier-Stokes/PAN AIR Analysis of a Nozzle and Plume Using an Irregular-Shaped Plume



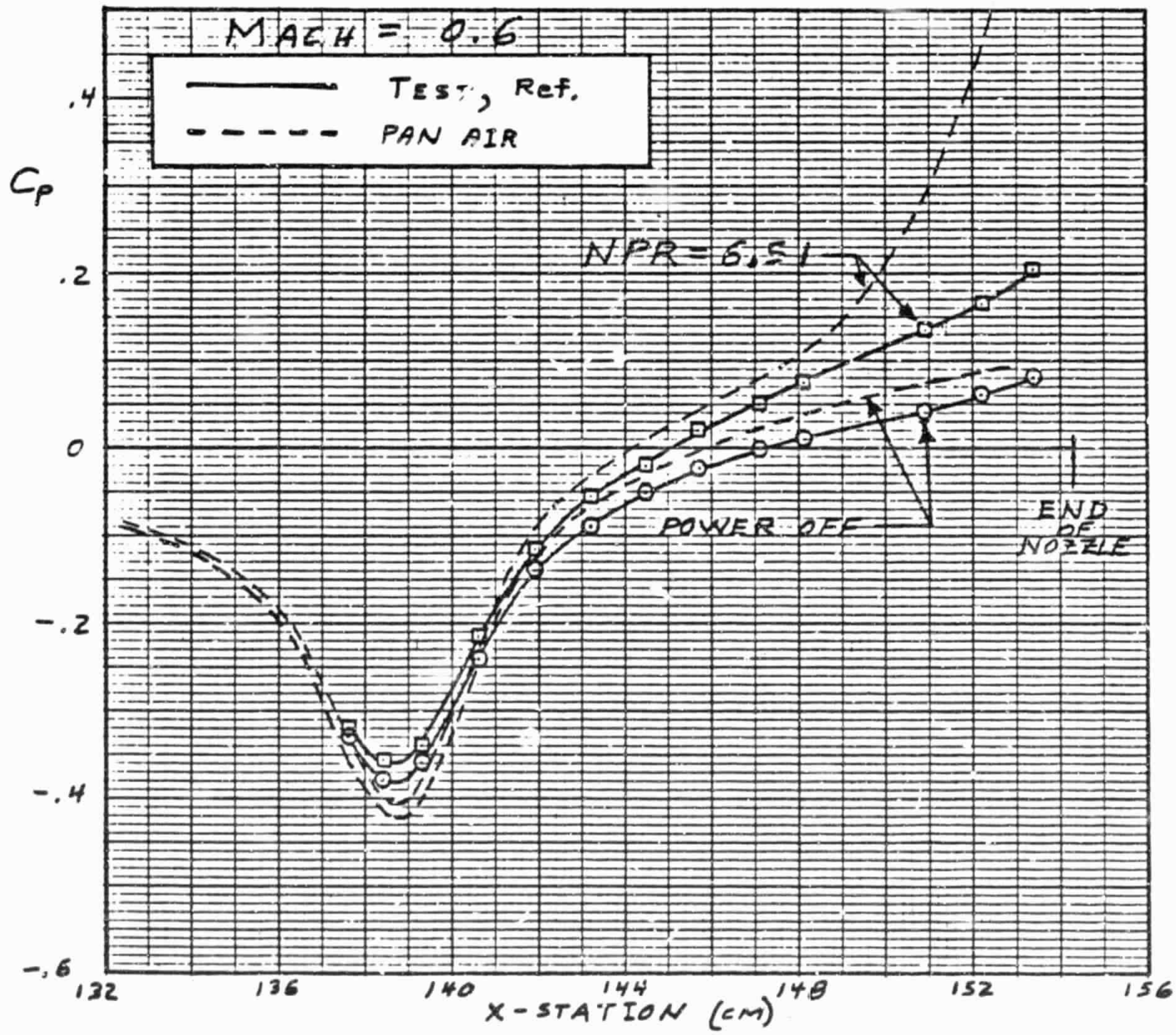


Figure 5-5 A Navier-Stokes/PAN AIR Analysis of a Nozzle and Plume Using a Smoothed Plume

boundary of the plume by a set of equations that relate the velocity at the location of the inviscid plume to the known velocities at the equivalent plume boundary.

The only data that are passed from the modified RAXJET code to the Engine Simulation Module are the inviscid plume boundary and the corresponding inflow velocities. These values are nondimensionalized by dividing by the exhaust exit radius and freestream velocity, respectively. Therefore, these parameters can be applied to a PAN AIR run by simply scaling them according to the dimensions of the configuration under study and the freestream velocity.

5.4 INTERFACE WITH THE AERODYNAMIC CODE

The PAN AIR code is quite complex and difficult to modify, however, it is extremely flexible in regard to the network shapes and boundary conditions that it will accept as input. Because of this flexibility, it was possible to develop the Engine Simulation Module such that it can interact with PAN AIR exclusively through the input data. To facilitate usage of the Engine Simulation Module, the RAXJET code has been directly included in the module as a group of subroutines. This eliminates the problems associated with passing data files between the modified RAXJET code and subroutines developed for other tasks to be carried out in the Engine Simulation Module.

User convenience has been emphasized during the development of this module. It is currently installed on the NASA Ames' VAX 11/780 computer system and may be executed with only two input files: (1) a basic PAN AIR input file and (2) an input file to specify the location of the nozzle and properties of the exhaust flow. These files must be in the same directory as the Engine Simulation Module as described in Appendix B.

An overview of the activities of the Engine Simulation Module is shown in Figure 5-6 and they are performed in the following order:

1. Read input data (i.e. name of original PAN AIR input file and properties of the jet exhaust).

2. Read original PAN AIR file.

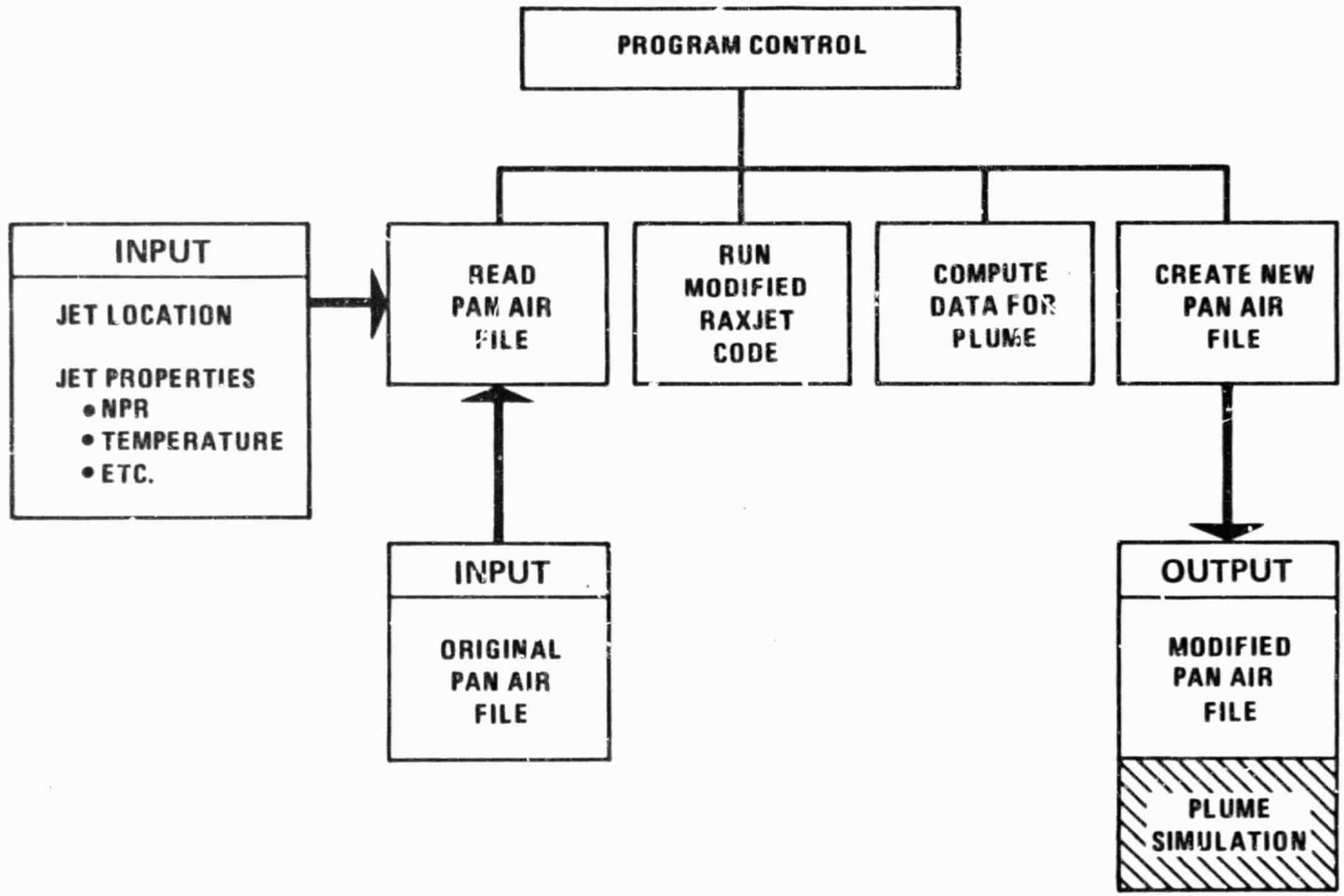


Figure 5-6 Overview of the Engine Simulation Module

3. Run modified RAXJET code.

4. Compute the panel corner-point coordinates and boundary conditions for the plume.

5. Compute the coordinates for the plume base and the plume wake.

6. Generate a modified PAN AIR input file including the plume simulation.

The output file generated by the Engine Simulation Module can be submitted for a PAN AIR run after a few items are manually checked as explained in Appendix B.

In order to make use of the two-dimensional data from the RAXJET code, it was necessary to make two assumptions regarding the three-dimensional behavior of plumes. The first assumption is that if the jet is injected at a small angle to the freestream flow then its center line will become parallel to the freestream at a distance equivalent to 10 nozzle exit diameters downstream of the exit as illustrated in Figure 5-7. The path of the plume centerline is assumed to be represented by the second order equation:

$$\Delta Z = A\Delta X^2 + B\Delta X + C$$

where the coefficients A, B and C are determined by applying the following three boundary conditions.

1. The nozzle centroid is a point on the path.
2. The path is normal to the average nozzle exit plane.
3. The path is parallel to freestream at 10 nozzle diameters downstream of the exit.

The second assumption regarding the three-dimensional behavior of plumes (illustrated in Figure 5-8) is that the plume expands radially outwards from the plume centerline path by the same ratio in which the two-dimensional plume expands. This assumption ignores three-dimensional effects on the plume boundary.

The generation of the plume networks is governed by the information provided through the input file. The user must specify which of the configuration networks abut the plume and also which edge of the network forms the abutment. Nozzle-type networks, shown in Figure 5-9, illustrate the positions of the solid panels to which the

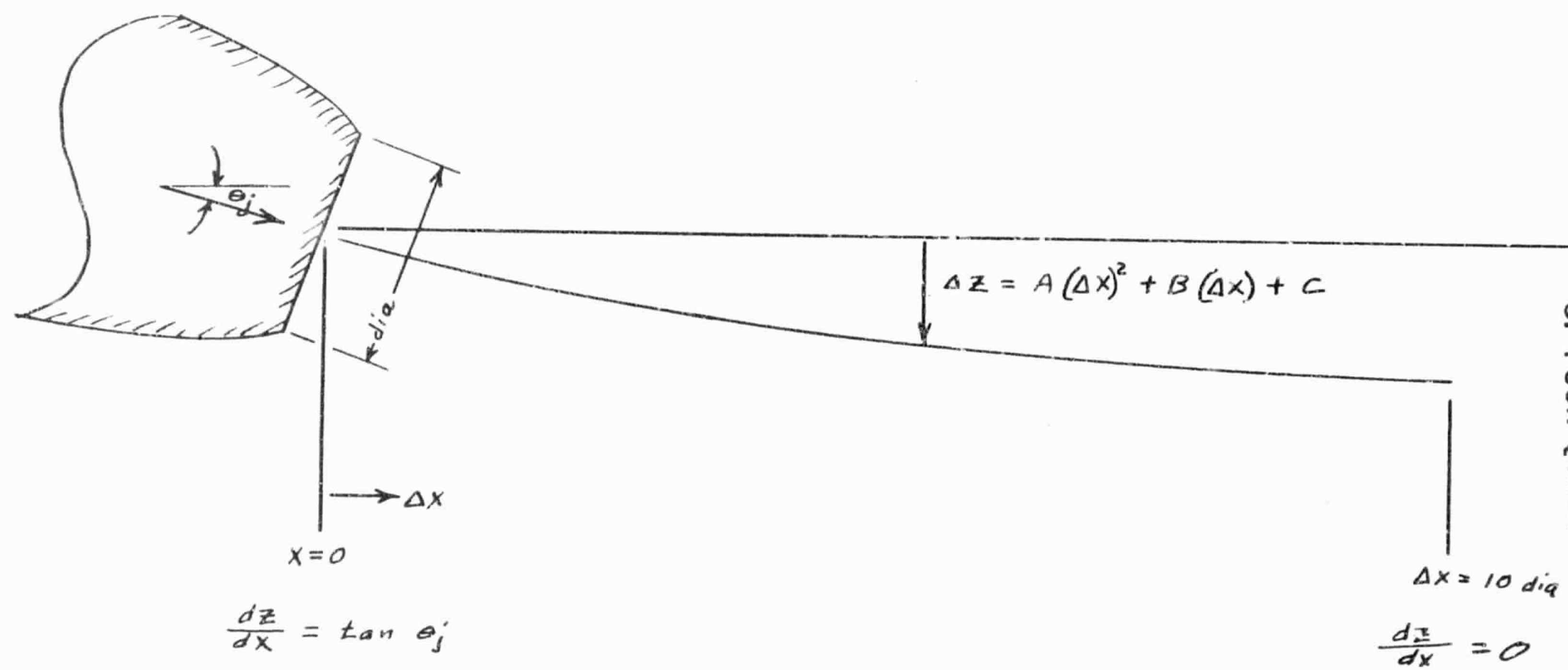


Figure 5-7 Centerline Path for Deflected Plumes

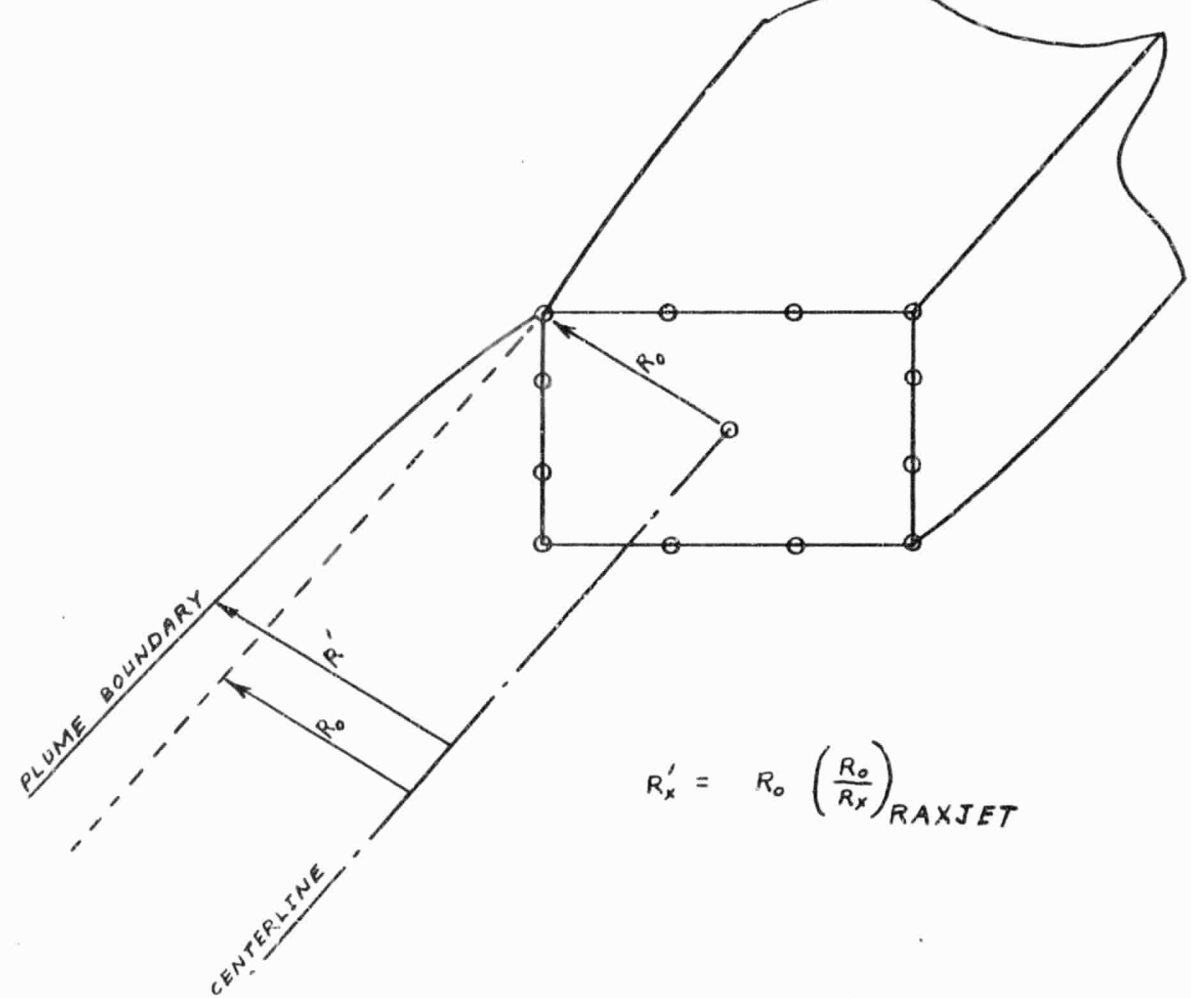


Figure 5-8 Expansion Method for Three-Dimensional Plumes

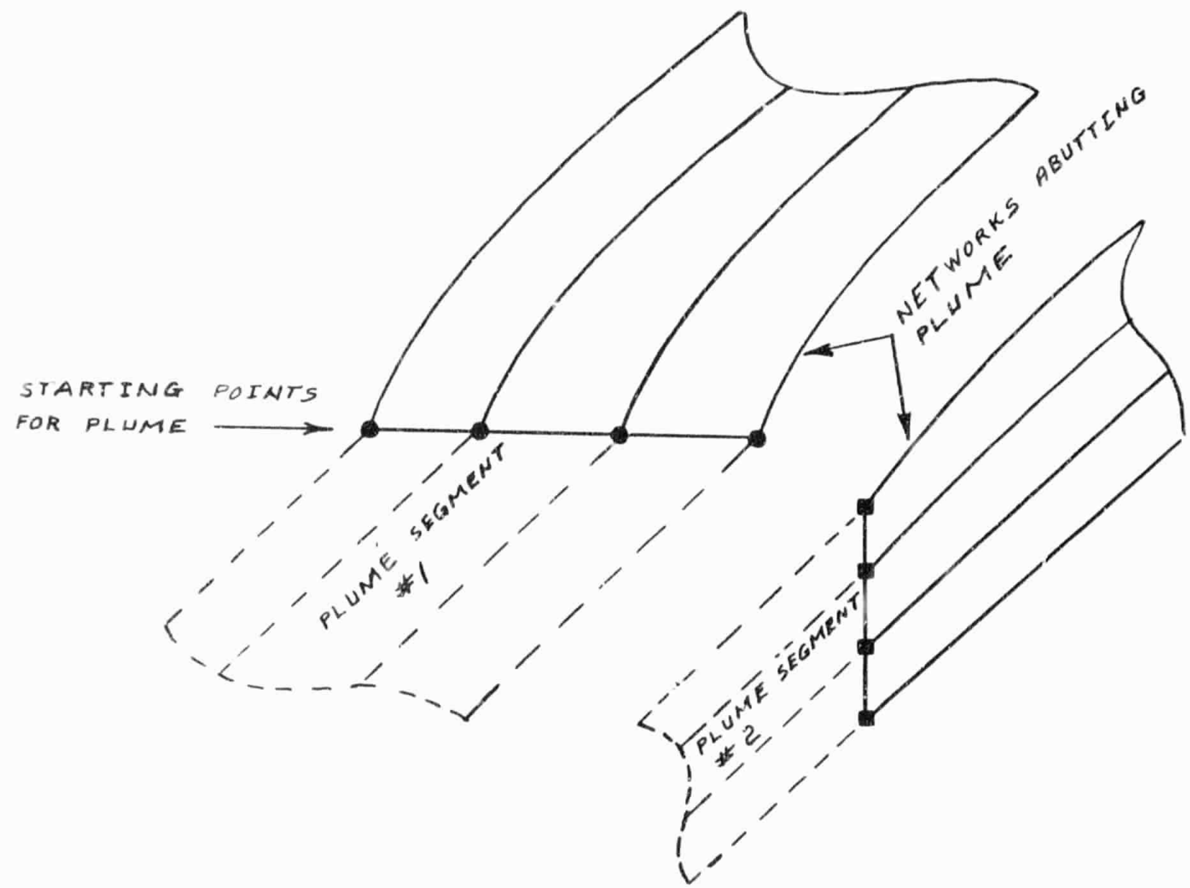


Figure 5-9 **Illustration of Abutment Between the Nozzle and Plume Networks**

plume networks will abut. One plume network is developed for each of the solid networks in the exit abutment. Thus, a plume may be comprised of several networks, such that each segment is constructed with exact corner-point matching at the abutment between the configuration and the plume networks. This type of abutment minimizes the likelihood of an error being encountered in the PAN AIR code due to a panelling deficiency.

PAN AIR's CLASS 2 boundary conditions are applied to the networks of the plume. This class is for a specified normal mass flux analysis and are applied through the equations

$$\sigma = -\vec{U}_o \cdot \vec{n} + \beta_{n_1}$$
$$\phi_L = 0$$

Specification of these boundary conditions allows the plume entrainment characteristics to be specified through the β_{n_1} term. The boundary condition information is automatically generated by the Engine Simulation Module, using the RAXJET inflow velocity at each panel center point, and included in the new PAN AIR file.

When the plume networks are added to the configuration panelling arrangement, the original nozzle exit network is automatically removed from the file, and base networks are added to the downstream boundary of the plume in a manner similar to that shown in Figure 5-10. One base network is developed for each of the networks defining the plume, and they all connect to a point at the plume centerline. The plume base network is also constructed to provide exact corner point matching at its abutment with the plume. PAN AIR's CLASS 4 boundary conditions are applied to the plume base network, which allows for the selection of specific terms. The governing equations that are used are

$$\mu = -\vec{U}_\infty \cdot \vec{\psi}$$
$$\phi_L = 0$$

These equations, in effect, specify that there is no flow tangent to the plume base and allow the flow normal to the base to be determined by the aerodynamic solution. Thus, the flow streamlines near the plume base do not have to turn a corner and follow the closure.

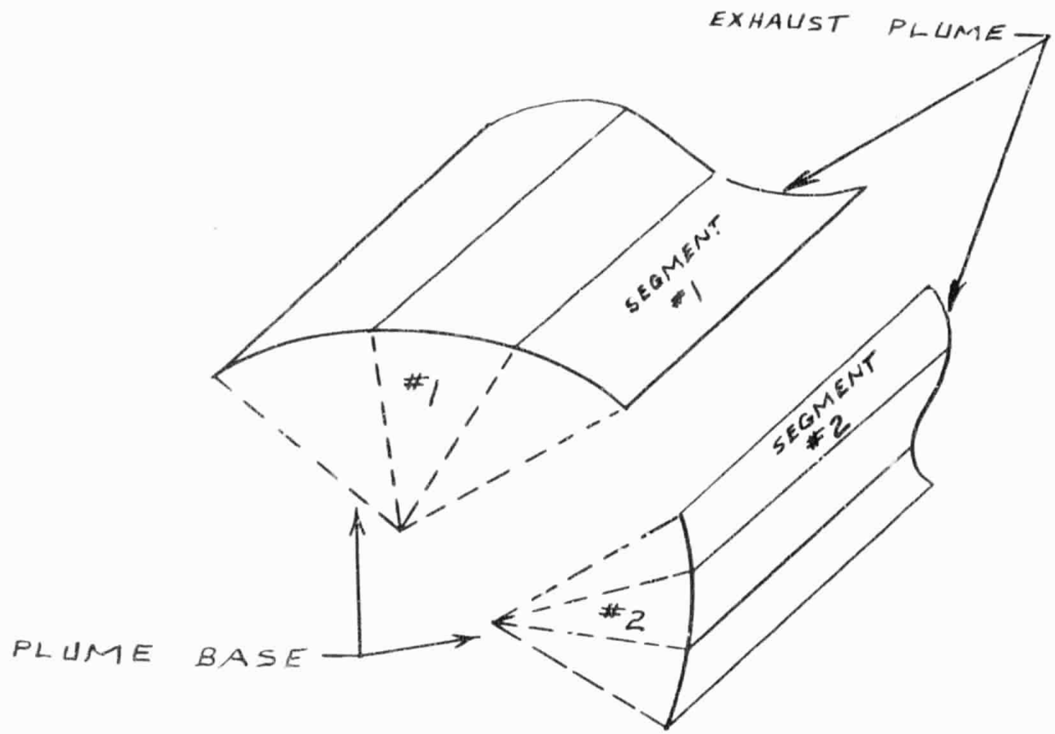


Figure 5-10 Plume-Base Networks

5.5 VALIDATION OF THE ENGINE SIMULATION MODULE

Isolated Nacelle Model. A preliminary application of the Engine Simulation Module was made by running the code to predict the effects of nozzle pressure ratio on the isolated nacelle model shown in Figure 5-11. This model was tested in the 16-Ft. Transonic Wind Tunnel at the Langley Research Center with high pressure air ejected through the nozzle to simulate a jet exhaust flow. Nozzle boattail pressure data were recorded at several nozzle pressure ratios and are documented in Reference 43. The measured effects of the nozzle pressure ratio on the boattail surface pressures are shown by the data in Figure 5-12. The lowest Mach number tested, 0.6, was used in this study.

The panelling arrangement shown in Figure 5-13 was developed to represent the basic geometry of the isolated nacelle model, and a PAN AIR file was prepared for a power-off run. The Engine Simulation Module was then exercised using this PAN AIR file to create three new files, each for a different nozzle pressure ratio. The plume radii distributions and the entrainment velocities are shown in Figure 5-14. PAN AIR runs were then made with each of the modified files, resulting in the boattail pressures shown in Figure 5-15. Comparisons with the test data (Figure 5-12) show that the predicted power effects have the correct trends but the pressure values near the nozzle exit are too large in magnitude.

To investigate the reason for this pressure discrepancy, the unmodified RAXJET code was run for a two-dimensional shape that was the same as a cross-section of the isolated nacelle model. Although RAXJET is a two-dimensional code, it has been shown in Reference 32 to be capable of handling the axisymmetric configurations, such as this model. The code was run for twenty iterations in order to allow the solution to converge and account for the viscous effects on the configuration as well as on the exhaust flow. The predicted boattail pressures, Figure 5-16, compare well with both the levels and the trends of the experimental data (compare Figures 5-12 and 5-16). This indicates that the information obtained from the RAXJET code is correct.

In the Engine Simulation Module, the boundary layer buildup and flow separation on the nozzle boattail are ignored. This should increase the blockage effects of the plume and hence, increase the boattail pressures just forward of the plume. In the absence of a boundary layer, the flow near the surface of the model must turn sharply in order to

(All linear dimensions are in centimeters unless otherwise noted.)

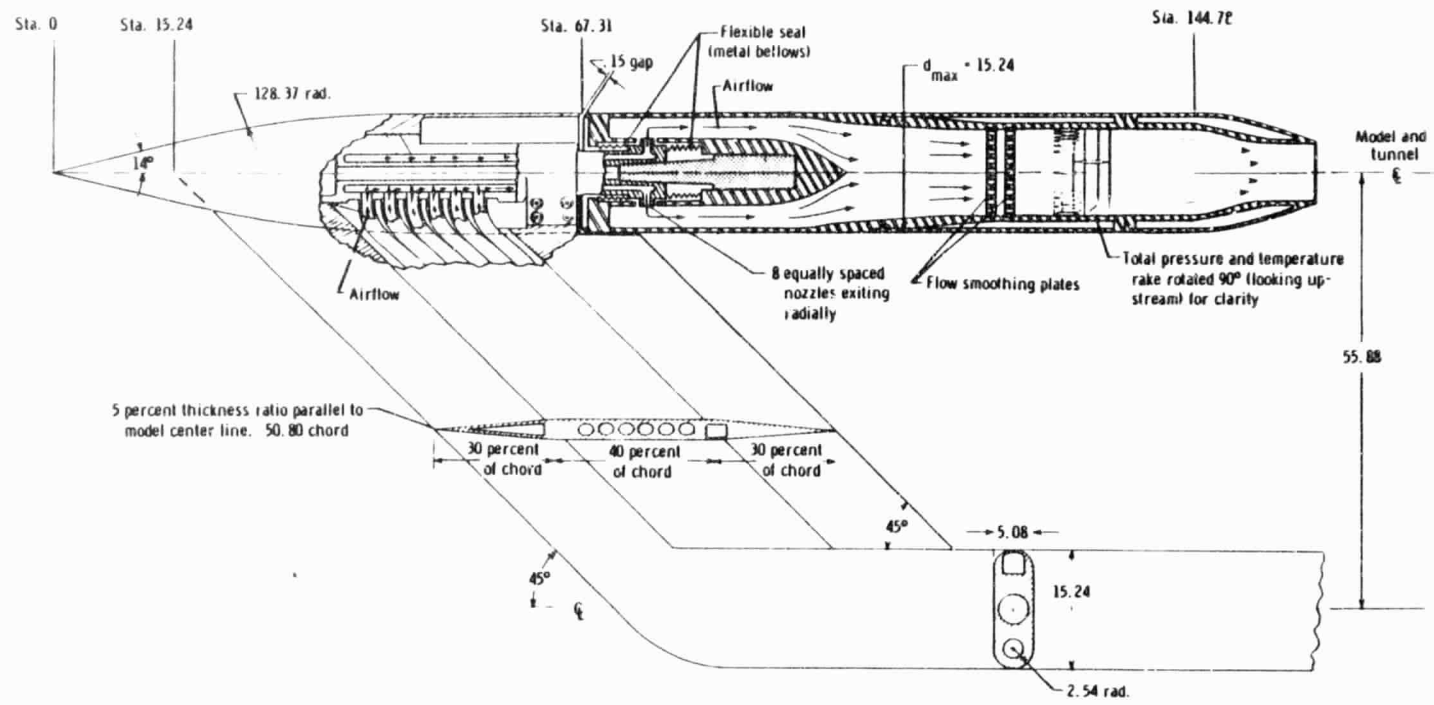


Figure 5-11 Isolated Nacelle Model

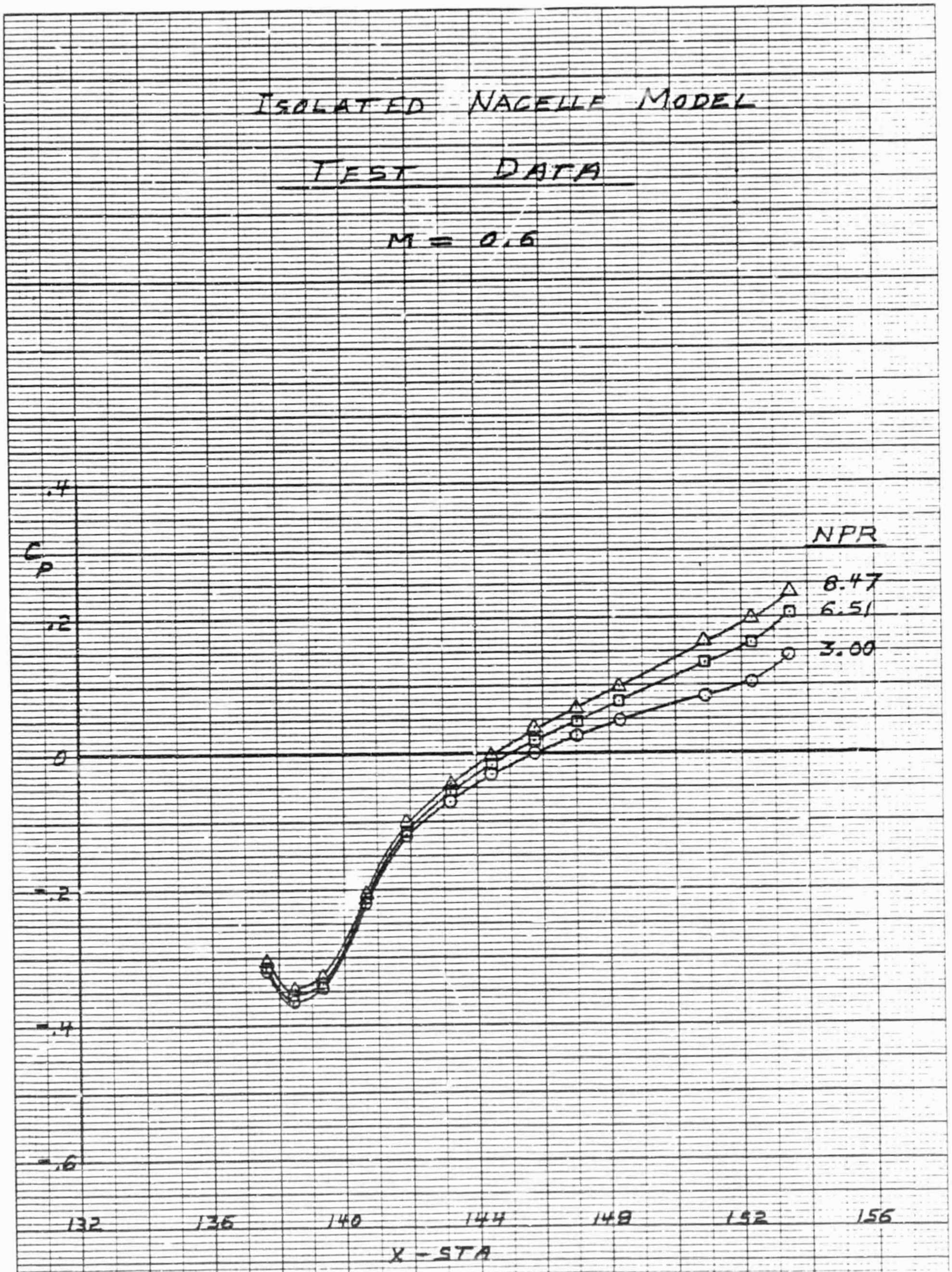


Figure 5-12 Nozzle Boattail Pressures Obtained from Experimental Tests

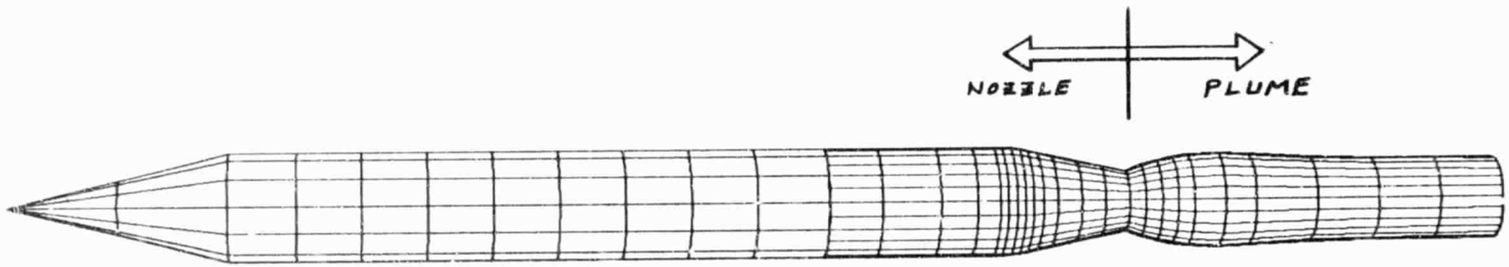
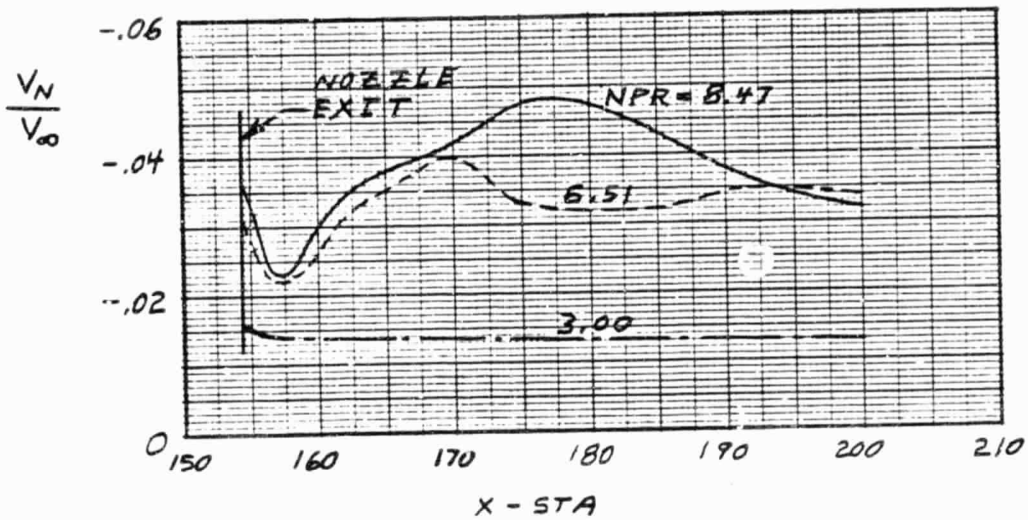


Figure 5-13 Panelling Arrangement for the Isclated Nacelle Model



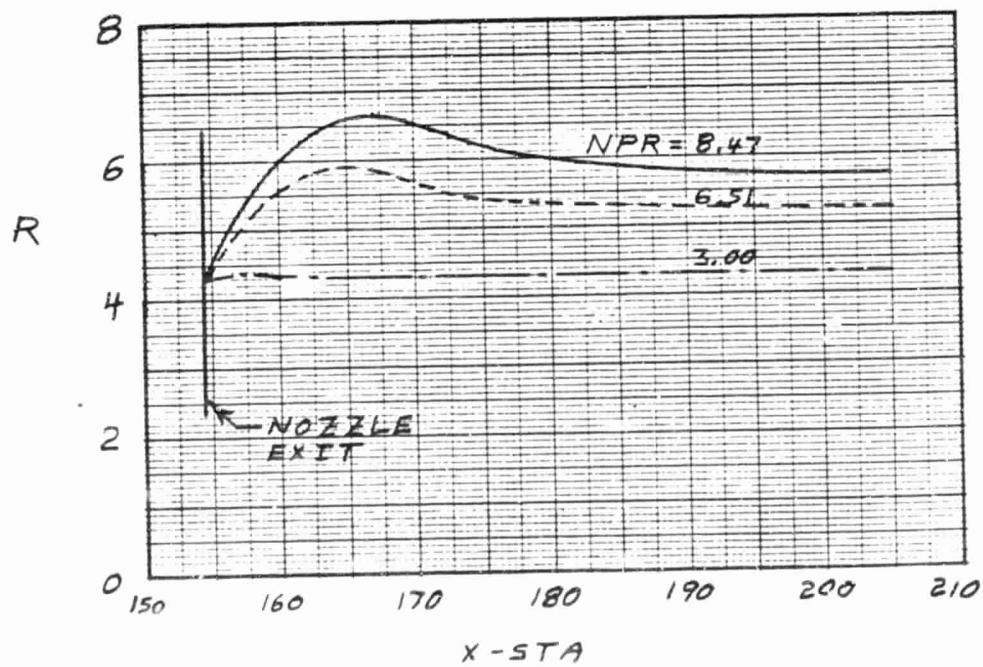


Figure 5-14 Plume Shapes and Entrainment Factors for the Isolated Nacelle Model

ORIGINAL PAGE IS
OF POOR QUALITY

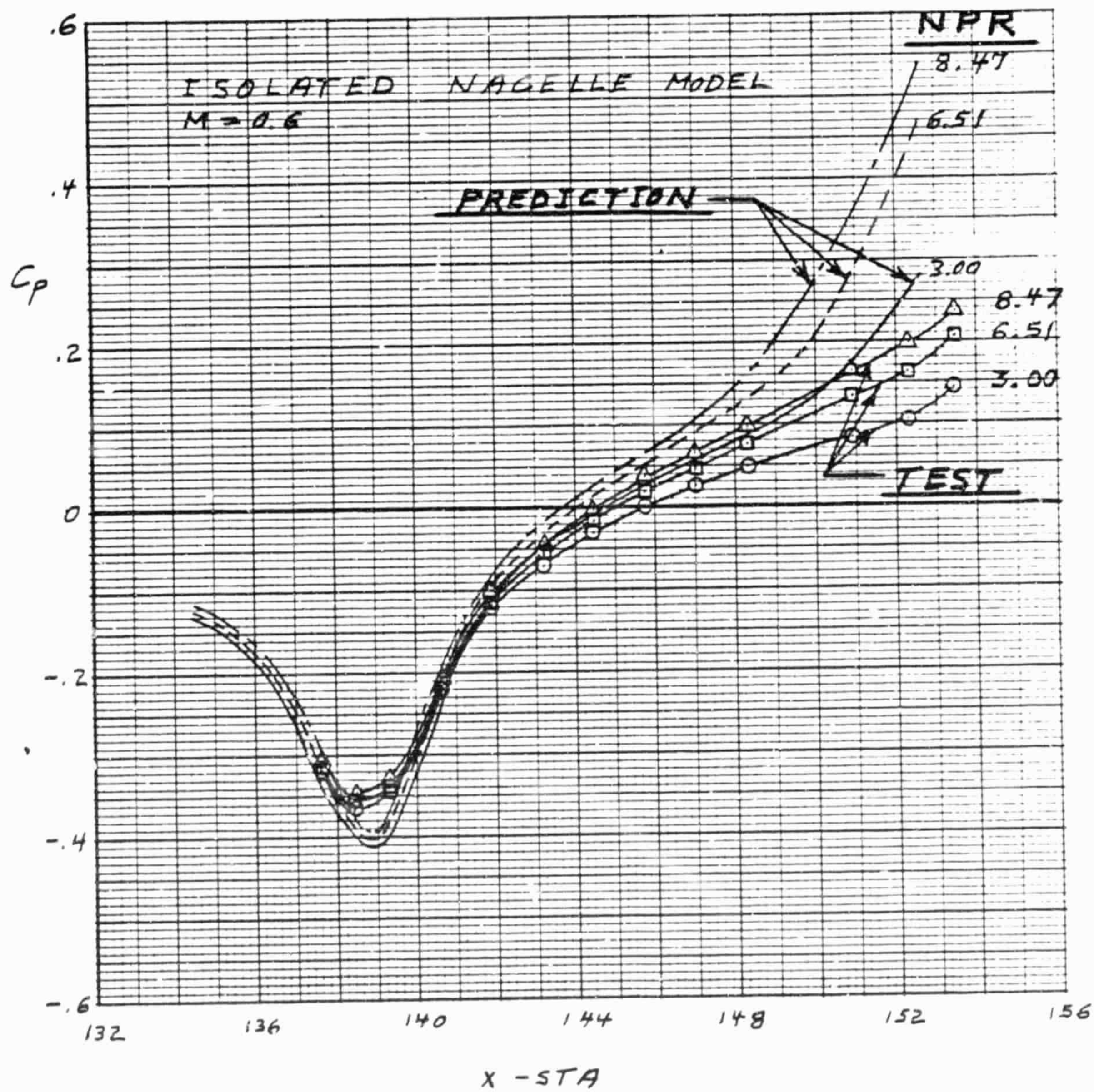


Figure 5-15 Nozzle Boattail Pressures Obtained from the Engine Simulation Module and PAN AIR

ORIGINAL PAGE IS
OF POOR QUALITY

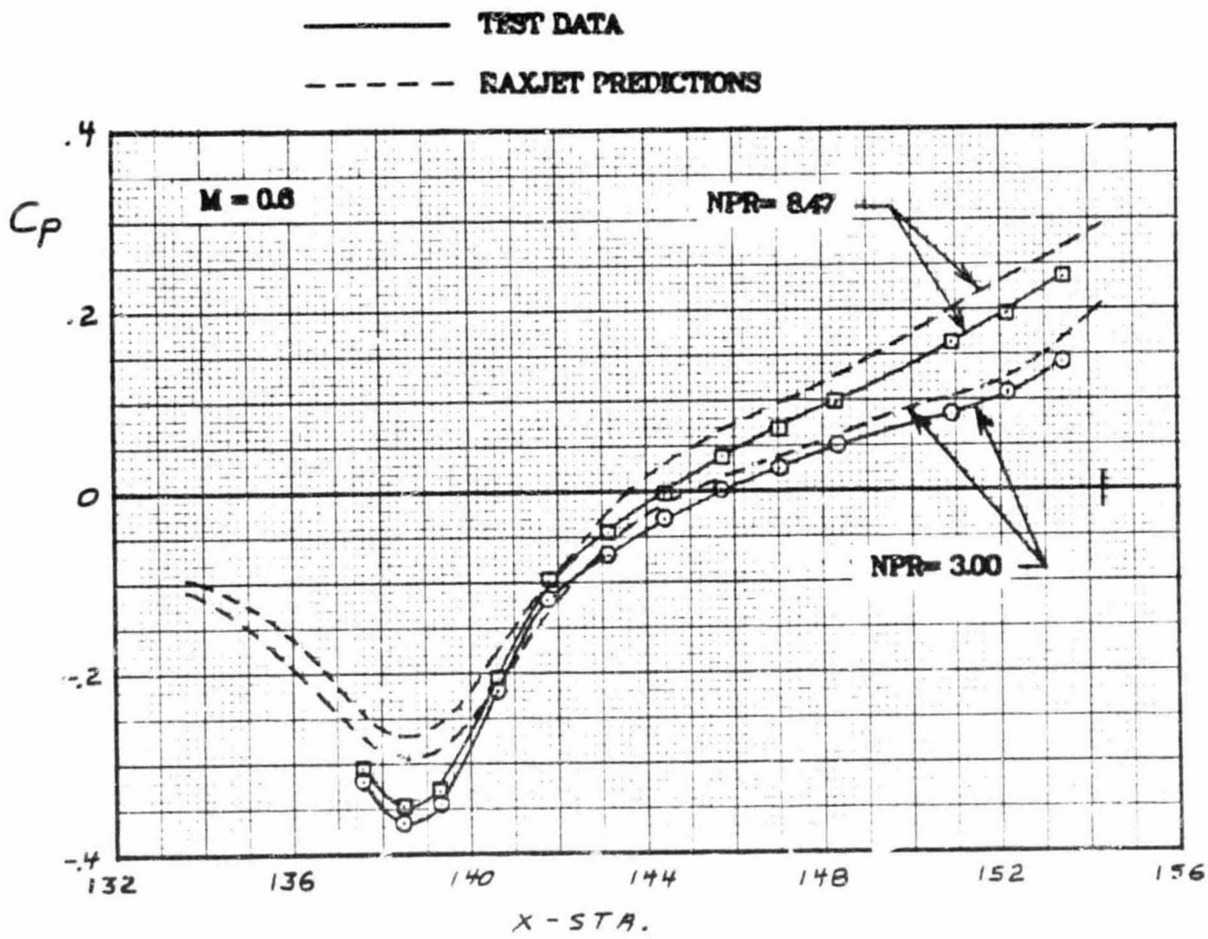


Figure 5-16 Nozzle Boattail Pressures Obtained from the Unmodified RAXJET Code

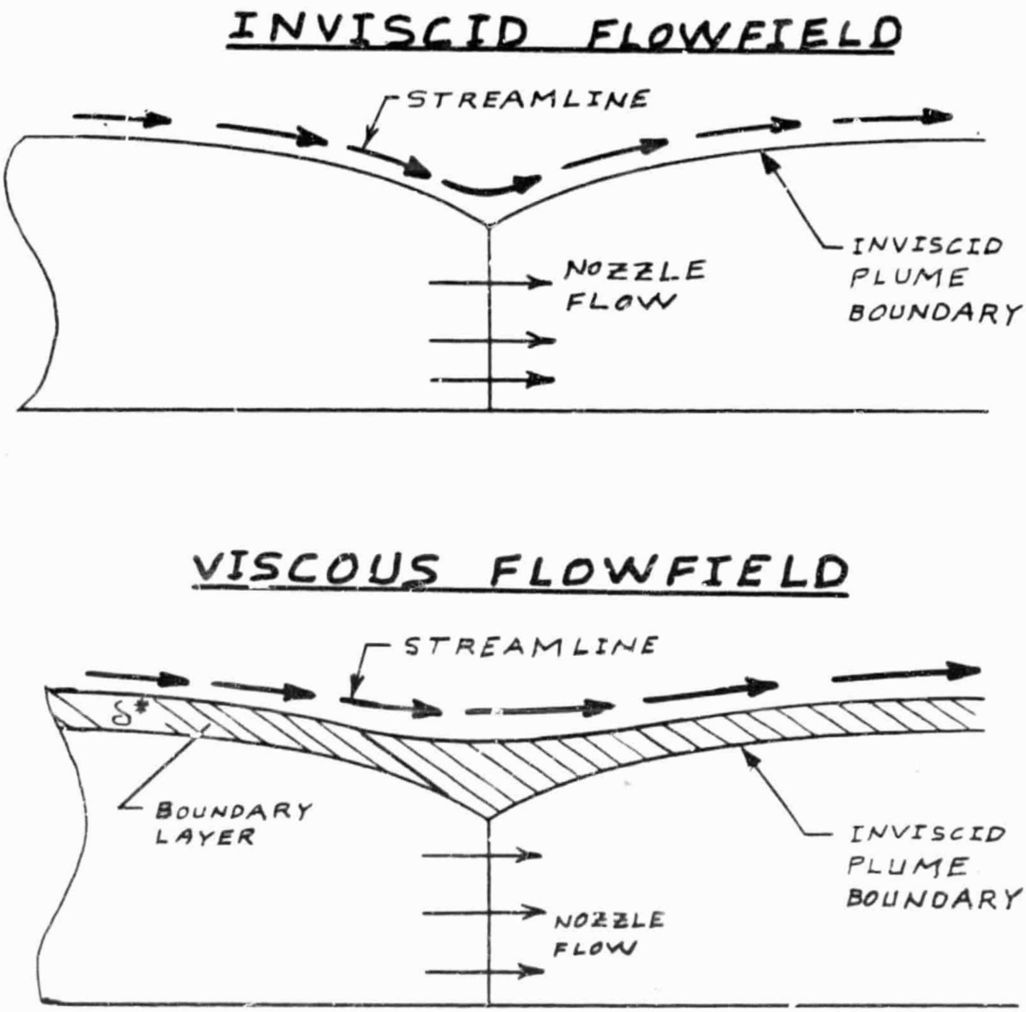


Figure 5-17 Effect of the Nozzle Boundary Layer on the Boattail Pressures

follow the surface of the plume, thus causing a compression (Figure 5-17). If the boundary layer were represented by a displacement thickness, as shown by the cross hatched area, the external flow would not be turned as sharply, and the boattail pressures would be reduced. It is anticipated that in future correlations at lower Mach numbers, the smaller effect of the boundary layer will be masked by the much larger effect of flow entrainment into the plume. If this proves to be the case, then power effects on V/STOL configurations will be obtainable at low speeds with the current approach. The use of this method at higher Mach numbers will probably require a viscous simulation on the nozzle boattail.

Ames/Vought Model. This model, discussed in Section 3.2, was used to validate the methodology developed in the Engine Simulation Module. Since this model was not tested with operating engines, typical values for nozzle exhaust parameters were used as input to the module. The modified panelling arrangement, with the plume added, is shown in Figure 5-18. This can be compared with the panelling arrangement for the basic configuration shown in Figure 3-11.

The file generated by the Engine Simulation Module, including the plume simulation, was analyzed with the PAN AIR code. This analysis was performed to check the ability of the Engine Simulation Module to communicate with the PAN AIR code and to establish general trends predicted for the effects of the plume. There are two factors that prevented a realistic analysis of this configuration with the existing code. The first is the large base area between the periphery of the plume and the edge of the aft fuselage. In the absence of a technique to accurately model this geometry, the nozzle base networks were assigned impermeable-surface boundary conditions. Thus, the flow was forced to follow the contours of the fuselage and base networks, which caused a turning of the flow through 90 deg at the nozzle base.

The second problem is the spacing of the exhaust plumes. The Engine Simulation Module is designed to perform analyses of single-nozzle configurations only. It may also be used for multi-nozzle configurations if the spacing is sufficient to justify the omission of mutual interference effects between the two exhaust flows. The two jets on the Ames/Vought model are closely spaced and undoubtedly have significant interference effects.

Although this model presents the two problems discussed above, an analysis was performed to determine if the correct trends would be predicted. Test data for this

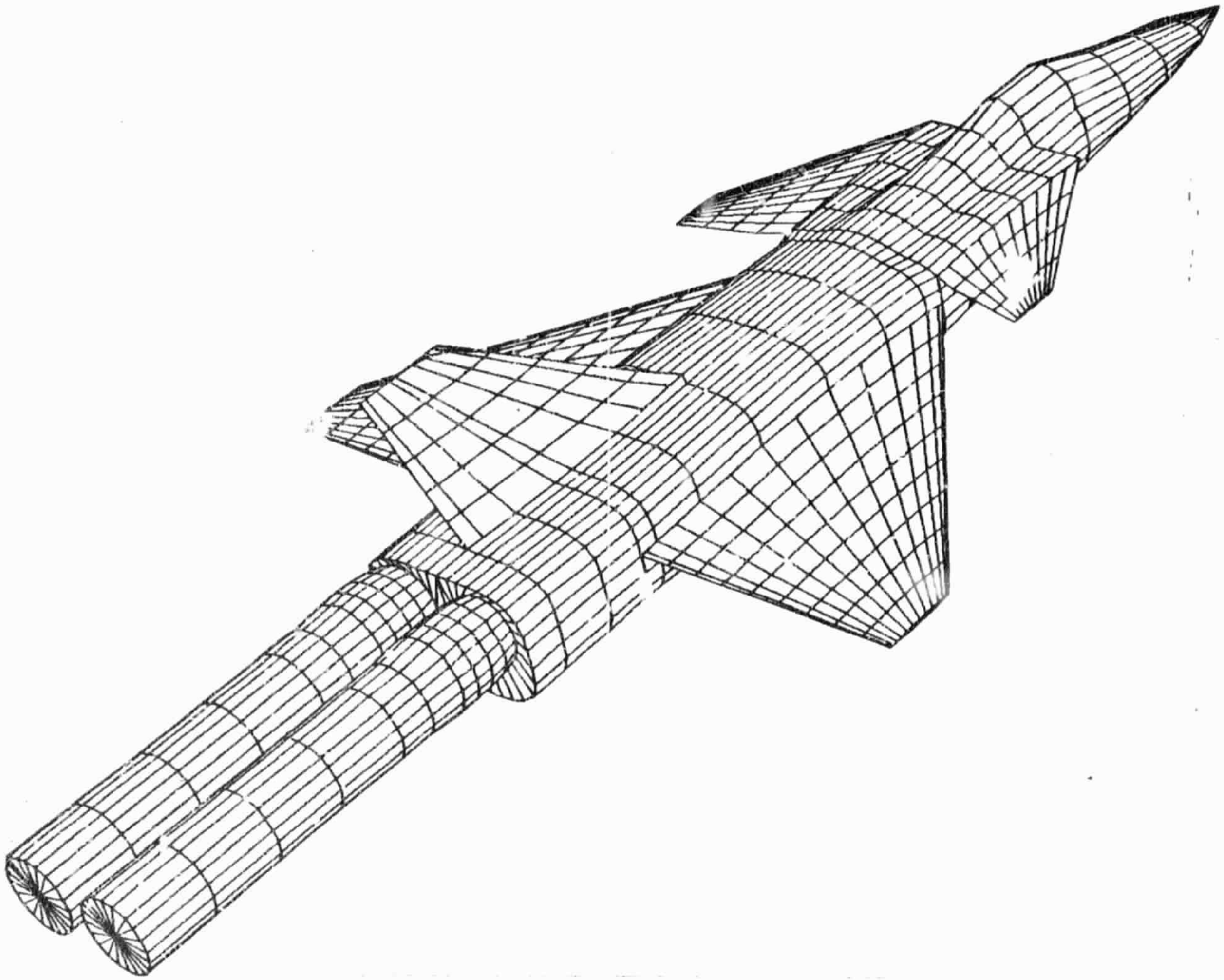
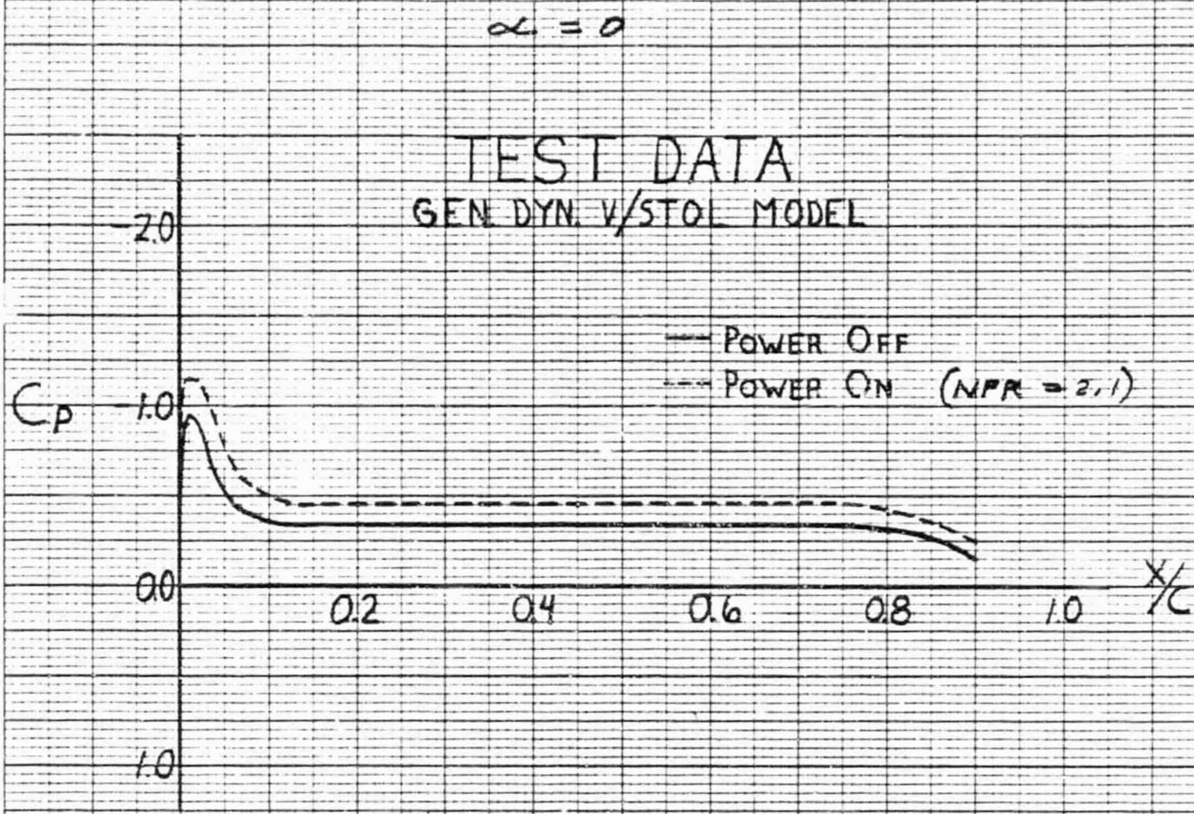


Figure 5-18 Panelling Arrangement of the Ames/Vought Model with the Plume Networks

model are available only with a small amount of nozzle exit flow, which resulted from a simulation of inlet conditions. This did not simulate realistic nozzle flow; therefore, the computed effects of the plume are compared with test data on the Ames/GD model (Figure 5-19). This qualitative test-to-theory comparison is for the wing upper-surface pressure near the one-third exposed semi-span of each model. An angle-of-attack of zero deg was selected to minimize the vortex effects in the experimental data. The power-effects in both sets of data are shown to propagate forward on the configuration and cause an almost constant pressure decrease over the wing chord.



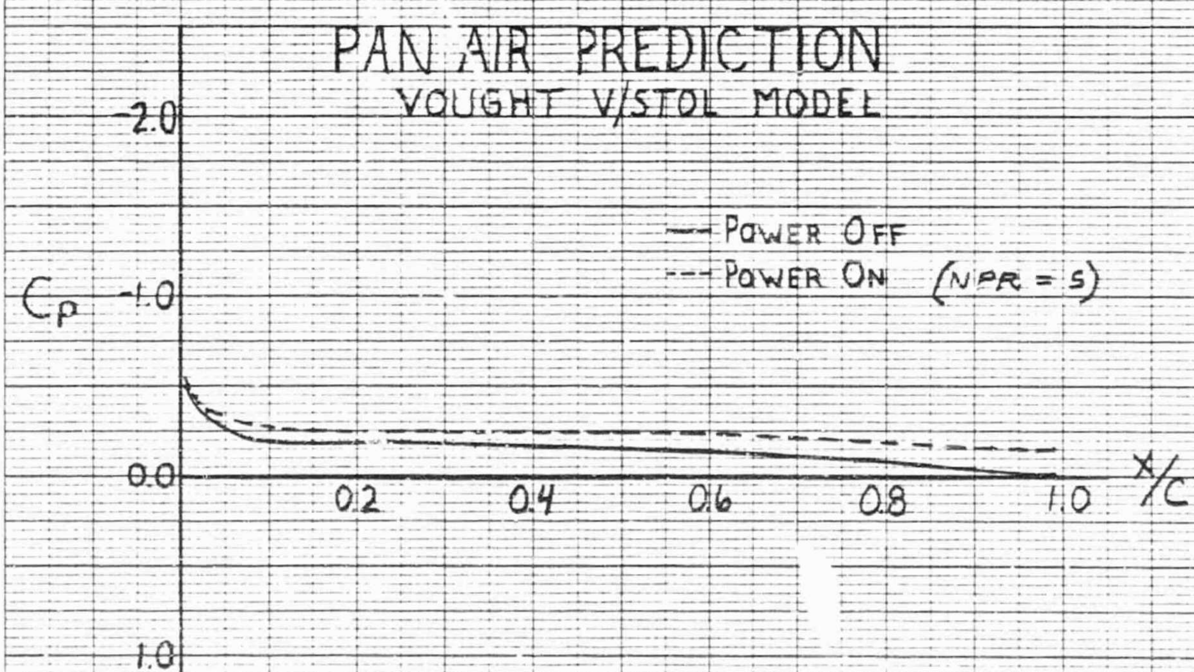


Figure 5-19 Effect of Exhaust Jet on Wing Pressures

6. VISCOUS SIMULATION MODULE

The simulation of viscous flow phenomena in a potential flow analysis is a difficult and tedious task, but one that needs to be addressed if realistic aerodynamic predictions are to be generated for V/STOL configurations. There are several aspects to viscous flow phenomena, namely,

- Laminar and turbulent boundary layers
- Laminar and turbulent flow separation
- Flow bubbles
- Vortex separation
- Trailing vortex sheet roll-up.

Each of these aspects need to be addressed in a comprehensive aerodynamic prediction method, but such is not entirely within the scope of this study. As an initial step toward the development of a viscous method, this study addresses two of the more prominent of the viscous flow phenomena - turbulent boundary layer buildup and trailing-edge separation on lifting bodies.

The classical way of simulating the viscous phenomena is to define an equivalent body that will simulate viscous effects when analyzed in potential flow. Definition of the equivalent body is usually accomplished through an iterative coupling of the analysis code with a boundary layer code. The success of this type of solution depends on:

- The accuracy of the aerodynamic flow-field prediction method
- The accuracy of the method to compute boundary-layer displacement thickness, separation lines, etc.
- The accuracy of the equivalent body to model all important aspects of the real flow field

The outline of an iterative scheme to simulate viscous effects using an equivalent body approach is shown in Figure 6.1. Starting from an inviscid solution, boundary layer characteristics are computed. These characteristics are used to define an equivalent

ORIGINAL PAGE IS
OF POOR QUALITY

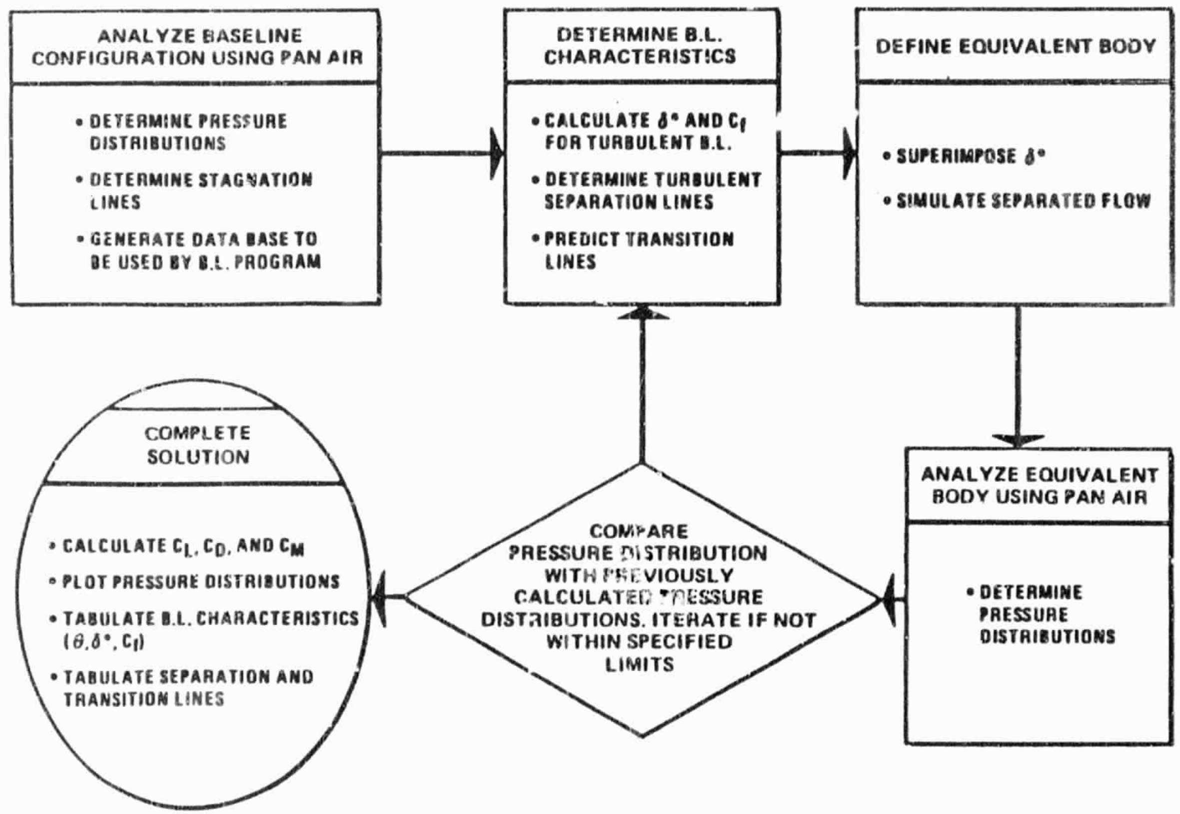


Figure 6.1 Iterative Solution for the Viscous Flow Simulator

body. An inviscid solution is obtained for the equivalent body. This process is repeated until the inviscid flow field computed for the equivalent body becomes invariant between successive iterations. At this stage, the solution is completed.

Due to the unexpected delays encountered in the availability of the baseline code, the programming of the complete iteration cycle has not been accomplished. The progress to date has been limited to developing methodology to define an equivalent body. The basic loop coupling the PAN AIR and Whitfield codes has been accomplished. This loop may be easily iterated as many times as desired by the user. The automation of this iteration can easily be completed when the development of a satisfactory viscous simulation is completed. The high cost of running PAN AIR and the manipulation of large blocks of flow properties and geometry for a realistic configuration has hindered the development of this iterative loop.

6.1 WHITFIELD BOUNDARY LAYER METHOD

The boundary layer prediction method selected for inclusion in the viscous simulation module is the 2-D Whitfield code (Reference 44). This method was selected because of the following reasons:

- o It has shown good agreement with experimental data.
- o The resulting computer code is relatively small.
- o Computation times are low.
- o It is widely used in industry.
- o It has been successfully applied as a strip boundary layer method in conjunction with a 3-D inviscid calculation method.
- o It is less complex thus more suited to inplement in an iterative-type solution.

Turbulent boundary layer methods can be classified as either integral or differential techniques depending upon the method of solution. Integral methods require the solution

to one or more ordinary differential equations; however, differential methods require the solution to a system of partial differential equations thus requiring more computational capabilities. Differential techniques are regarded in general to be more accurate than integral methods, but due to their requirement for more computer time and storage are not suited for the present application.

Whitfield attempted to develop an integral method that would rival the accuracy of differential methods without taxing the available computer resources. His method differs from other integral techniques primarily in the velocity profile and the velocity-temperature relationships used. The velocity profile used by the Whitfield method is an analytical expression that depends on the local values of skin friction, shape factor, and Reynolds number based on momentum thickness. The significance of this expression is that no new parameters are introduced into the integral equations.

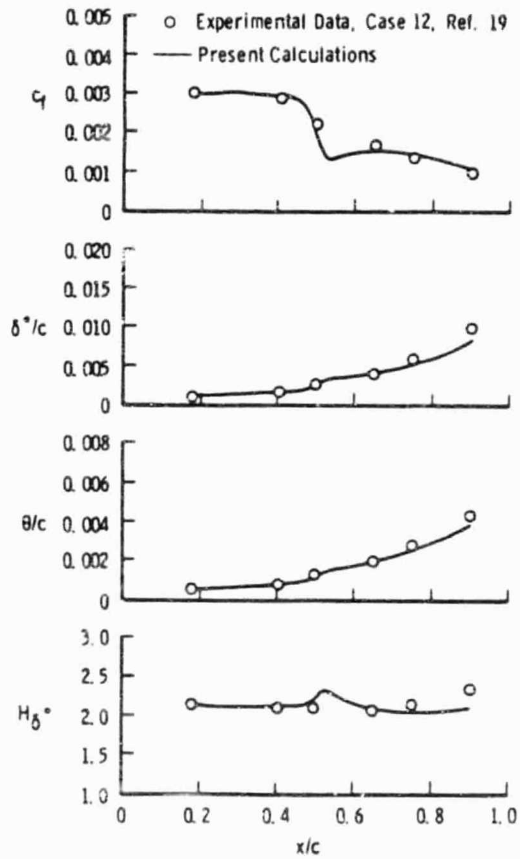
Based on the results presented in the literature, Whitfield achieved the desired results. Shown in Figure 6.2 taken from Reference 44, are the results presented by Whitfield for the boundary layer displacement thickness, momentum thickness, shape factor, and local skin friction coefficient for the upper surface of an airfoil. The velocity profiles on the same airfoil are shown for two values of Reynolds Number in Figure 6.3 which was also taken from Reference 44. The agreement with experimental data indicated in these and other comparisons was one of the major reasons for the selection of the Whitfield procedure.

6.2 INTERFACE WITH THE AERODYNAMIC CODE

The Viscous Simulation Module was formulated to apply a two-dimensional boundary layer procedure to three-dimensional flow fields using a strip approach on lifting surfaces. The chordwise pressure distributions computed by the PAN AIR procedure at several span stations are used as input to the Whitfield procedure. The resulting boundary layer predictions for displacement thickness and separation points are used to define an equivalent body which when analyzed in potential flow will simulate boundary-layer buildup and trailing-edge separation effects.

Two approaches defining the equivalent body were considered. The first was to add the calculated displacement thickness normal to the surface and thus define a new surface

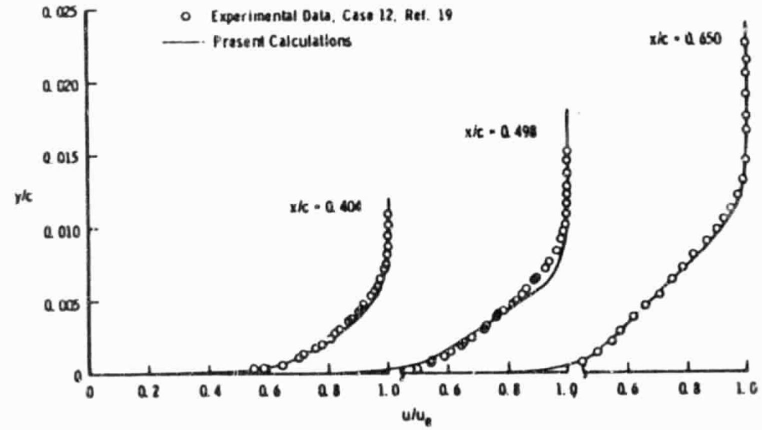
ORIGINAL PAGE IS
OF POOR QUALITY



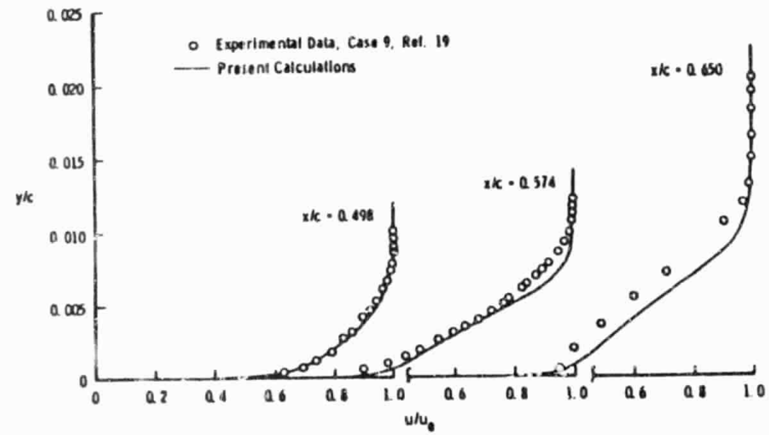
Upper surface boundary-layer properties of the RAE 2822 airfoil for $M_\infty = 0.730$, $Re_{\infty,c} = 2.7 \times 10^6$, and $\alpha = 3.19$ deg (Case 12).

Figure 6.2 Whitfield Results for Upper Surface of Airfoil (Taken from Reference 44)

ORIGINAL PAGE IS
OF POOR QUALITY



Velocity profiles in the shock region of the RAE 2822 airfoil for M_∞ = 0.730, $Re_{\infty,c}$ = 2.7 x 10^6, and α = 3.19 deg (Case 12).



Velocity profiles in the shock region of the RAE 2822 airfoil for M_∞ = 0.730, $Re_{\infty,c}$ = 6.5 x 10^6, and α = 3.19 deg (Case 9).

Figure 6.3 Whitfield Predictions of Velocity Profiles for Upper Surface of Airfoil (Taken from Reference 44)

coordinate for input into PAN AIR. This approach had the disadvantage that since the network geometry was being modified, the intersection abutment definitions between adjacent networks would be effected. This could consequently lead to a redefinition of a large part of the configuration geometry.

The second approach was to model the equivalent body by specifying a normal mass flux emitted by the network as illustrated in Figure 6.4. In this approach, the local mass flux is defined by the following equation.

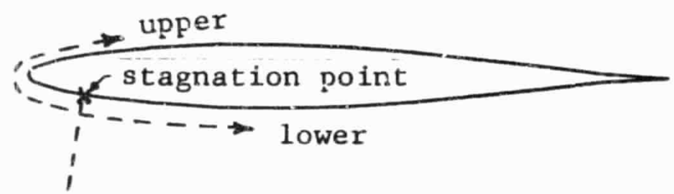
$$\beta_{nm} = \left[u_0 \frac{\partial \delta^*}{\partial s} \right]$$

This approach does not require any changes in geometry but does require a boundary condition specification change and an input of the mass flux values for the analysis of a new PAN AIR problem. This approach was selected because it was judged to be easier than the geometrical definition of the basic configuration.

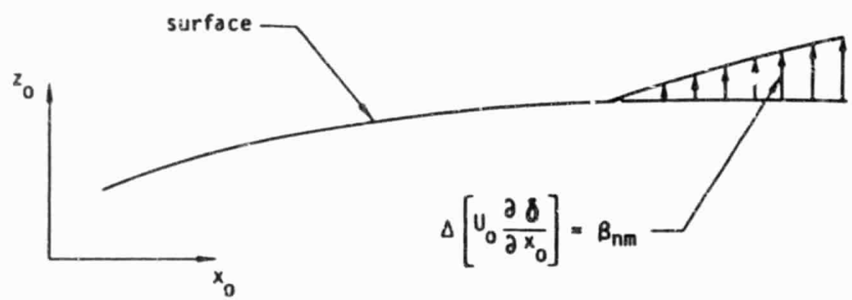
The loop required to implement this approach is depicted in Figure 6.5. The inviscid PAN AIR problem is first run without any corrections. The file containing the inviscid flow properties is generated on logical unit 10 by the PAN AIR code if the input contains the required records for executive Print-Plot Processor (PPP) module. These records are described in the PAN AIR Users Manual (Reference 16). The file created on logical unit 10 in the PPP module must be transfered to the VAX computer to execute the Viscous Simulation Module. The VAX system will allow for more user interaction and convenience. Once the flow properties are transferred to the VAX, they are used as input to the Whitfield Boundary Layer Procedure. The Whitfield predictions for displacement thickness and separation points are then used to calculate the mass flux required to define the equivalent body. A new PAN AIR problem is then generated. Once the inviscid flow properties are transferred to the VAX, this loop is completely automated. It is assumed that both the original PAN AIR problem and the calculated inviscid flow properties are stored in separate files on the VAX system.

The programs required to produce a new PAN AIR problem for the equivalent body are shown in Figure 6.6. All of these programs are executed by a central command file named VISC.COM on the VAX computer. The user is assumed to be interactive during this loop and is occasionally prompted for information such as file name containing flow properties and Mach number and Reynolds Number for boundary layer analysis. The source listing for each of these programs and user instructions for the execution of the

BOUNDARY LAYER DISPLACEMENT SIMULATION



LINEARIZED MODELING FOR BOUNDARY LAYER DISPLACEMENT EFFECT



BOUNDARY LAYER MODELED BY A SPECIFIED NORMAL MASS FLUX EMITTED BY THE AIRFOIL NETWORK

Figure 6.4 Equivalent Body Simulation

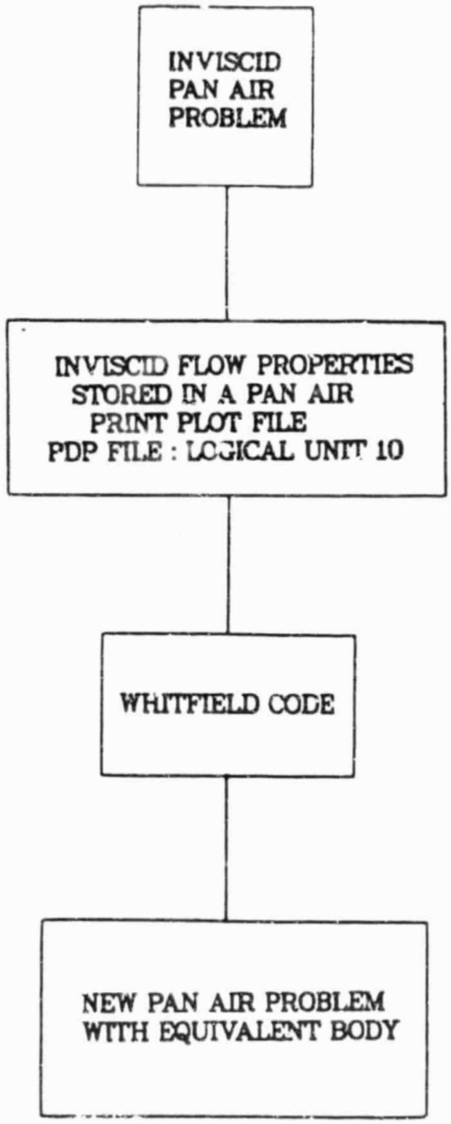


Figure 6.5 Loop Required For Viscous Simulation Module

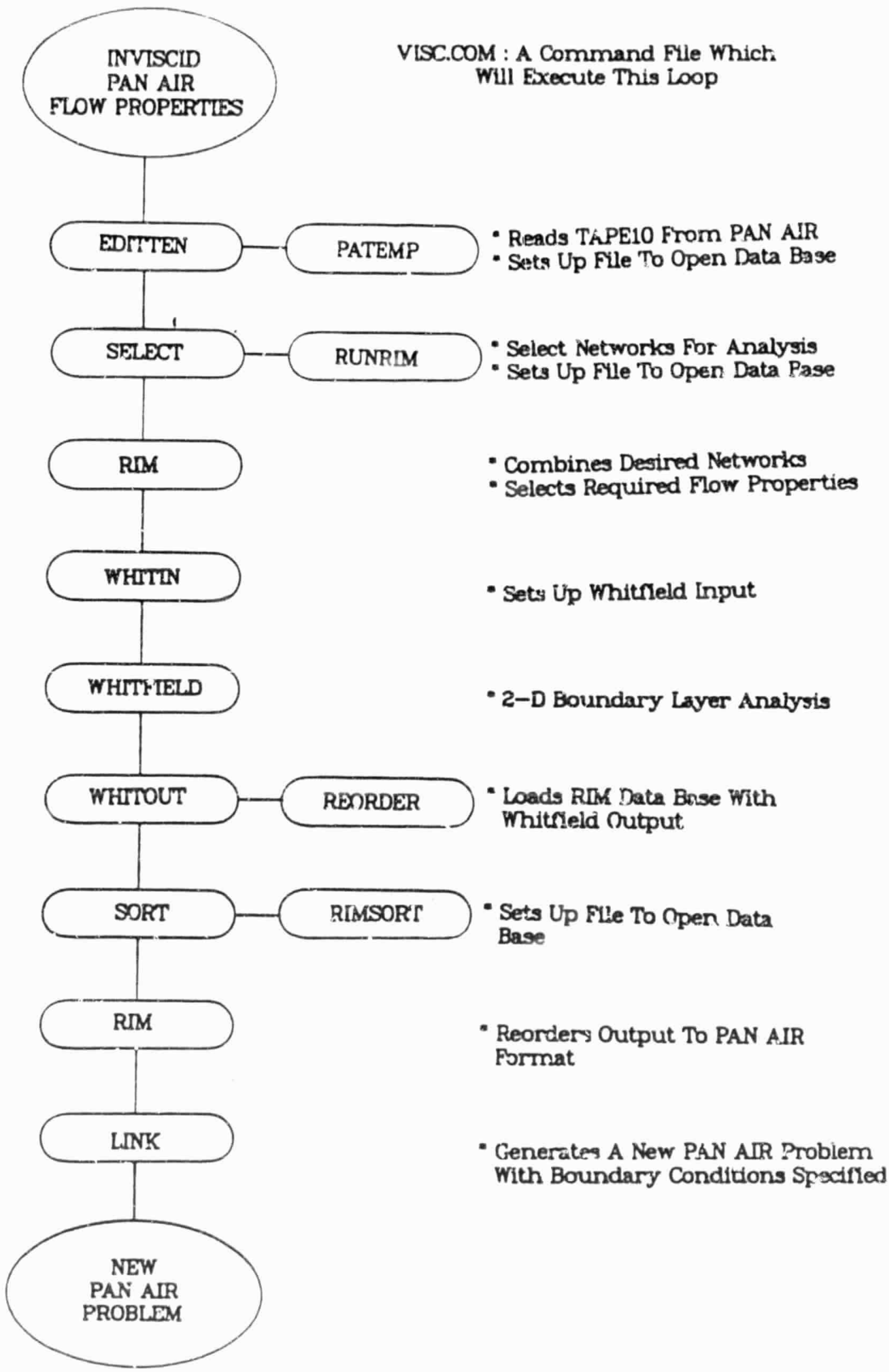


Figure 6.6 Programs Required to Link PAN AIR and Whitfield Procedures.

Viscous Simulation Module are included in Appendix C. A discussion of each of these programs including required input and primary function is contained in the following paragraphs.

The program which reads the file containing the inviscid flow properties from PAN AIR is named EDITTEN. This program was developed at NASA Ames. This program is the first program executed by the command module and accomplishes the following tasks:

- Reads the output file generated by the PPP module of the PAN AIR code and separates out the desired flow properties.

- Generates a VAX command file named PATEMP. COM that, when executed, sets up a RIM data base containing the inviscid flow properties.

- Generates a file named PATEMP.INFO containing information from the PAN AIR run which can be printed by the user if desired.

- Generates a file named INDEX.TMP which contains the indices of the networks in the PAN AIR problem.

The command file named PATEMP.COM generated by EDITTEN to set up the RIM data base containing the calculated flow properties is the second program executed by the command file, VISC.COM. This file contains the necessary RIM commands to establish a RIM data base which will have the name NETWORKS. The third program executed by the command module is the SELECT program. In many cases the upper or lower surface of the wing may be divided into several networks. The user must provide the number of networks and their identification numbers for both the upper and lower surfaces; these will be displayed on the terminal for the user. The SELECT program will generate a command file named RUNRIM.COM, which will utilize the previously generated data base file NETWORKS, select the required flow properties for only the designated upper and lower surface networks, and store them in two VAX files named WINGU.DAT and WINGL.DAT. The RUNRIM.COM containing the RIM commands to accomplish these tasks is executed next by the central command file, VISC.COM.

The files containing the required flow parameters, WINGU.DAT and WINGL. DAT, need to be converted to a format acceptable to the Whitfield procedure. The program

WHITIN was developed to accomplish this task. The user is asked to specify a geometric tolerance value to determine panel alignment in the chordwise direction, a Mach Number, and a Reynolds Number per million for the boundary layer analysis. A geometric tolerance value is required because the panel control points where the flow properties are computed by PAN AIR do not necessarily lie exactly on a straight line. The stagnation point at each station is then located. The location of the stagnation point is then used to generate two boundary layer analysis problems, one for the upper surface and the other for the lower surface. The length measured from the stagnation point along the surface is used as the length parameter in the boundary layer calculations. The result is a file named UPPER.DAT containing the boundary layer problems (one for each station) for the upper surface and a file for the lower surface names (LOWER.DAT). These files are then used as input to the Whitfield Procedure. A third file named STAG.DAT is also generated by the WHITIN program to keep track of the location of the selected stagnation points at each station.

The results from the Whitfield program are stored in two VAX files named UPROUT.DAT and LOWOUT.DAT. It is now necessary to sort the results back into an order consistent with the PAN AIR scheme, i.e. by network column number, and row number. The RIM program was again used as the basic tool for this task. A program named WHITOUT was formulated to utilize the Whitfield program output files to generate a command file named REORDER.COM for creating a data base usable by RIM. At the completion of the WHITOUT program, the RIM data base named ORDER is created using the REORDER.COM file.

Once the data base ORDER is established, the commands to access the data base, sort the data into the proper order, and store the results is needed. The program SORT was written for this task. This program produces a VAX command file named RIMSORT.COM containing the RIM instructions to accomplish the desired results. The viscous module executes this command file at the completion of the SORT program. The results are stored in the VAX files named UPPER.OUT and LOWER.OUT.

The only remaining task to complete the loop is to combine the boundary layer results with the original PAN AIR problem to generate a new problem with an equivalent body. This is accomplished in the program LINK. The boundary conditions needed to define the equivalent body are included in the new problem. An example is included in Appendix C.

The viscous effects module is designed to operate on only one lifting surface at a time. The lifting surface may be composed of several different networks on the upper and lower surfaces. If more than one lifting surface is to be analyzed, (for example, a wing and a canard), they must be processed through the viscous simulation module consecutively. After a new PAN AIR problem has been generated for the first lifting surface, it is used to generate another PAN AIR problem for the next lifting surface and so on. The final PAN AIR problem will have the boundary layer corrections for all lifting surfaces included.

6.3 RESULTS FOR VOUGHT V/STOL MODEL

The Vought VATOL Model was analyzed with the PAN AIR Code, as discussed in Section 3.2 and the aerodynamic flow properties were saved on a TAPE 10 file for use in the Viscous Simulation Module. The preliminary results of an analysis of the viscous effects of this model are presented in this section.

An angle of attack of 8 degrees was selected for the present analysis. The output file containing the PAN AIR predicted flow properties was input to the Viscous Simulation Module. Boundary layer calculations were made for both the wing and canard surfaces. A second PAN AIR run was made using the equivalent body to simulate the boundary layer effects as defined by this module. As previously discussed, the present equivalent body model does not result in an actual change to the configuration geometry but consists of a specified mass flux over the surface. This mass flux is proportional to the gradient of the boundary layer displacement thickness. The main purpose of this exercise was to verify the Viscous Simulation cycle and to begin the evolution of an equivalent body definition technique.

The transpiration method currently being used as an equivalent body appears to be the best approach to simulate viscous effects. Further refinements to the equivalent body will be necessary as additional data correlations are made. As the separation grows at higher angles of attack thus creating a large separated wake, the present equivalent body model may be altered to simulate these effects. Additional specifications of the boundary conditions on the separated wing and wake may be necessary to correctly model the viscous effects.

The effects of viscosity on the wing pressures of the Ames/Vought Model were computed with this technique and are shown as the increments between the solid and dashed lines in Figure 6.7. At both of these wing stations, the leading-edge suction pressures were decreased by the viscous effects. This is a well-known result of viscosity. There were other factors that overshadowed the viscous effects near the trailing-edge of the wing. As shown in Figure 6.7, the upper and lower wing surface pressures computed by PAN AIR divergered near the trailing edge at the 38% span station. Further outboard at the 63% span station, the pressures crossed near the 95% chordline and then diverged. Thus, the PAN AIR code did not satisfy the Kutta condition at the wing trailing edge.

These inviscid PAN AIR predictions were used as input to the Viscous Simulation Module as a preliminary checkout problem to verify that the data-handling routines in the module were operating correctly. The module performed the analysis and computed the boundary-layer thickness shown superimposed on the airfoil in Figure 6.7. Here the computed pressures on the upper surface did not show a strong favorable pressure gradient at the trailing edge. The computed boundary layer displacement thickness at this station shows the anticipated trends. The primary impact of this boundary layer is to reduce the effective camber of the wing. The resulting changes in the wing pressure coefficients are indicated by the pressure plots. An assessment of the validity of the pressure changes near the trailing edge must be deferred until the Kutta condition is satisfied in the PAN AIR analysis.

Several improvements to the equivalent body definition and boundary layer method are underway but were not utilized in the current analysis. The present equivalent body used the free stream velocity as a multiplication factor to the gradient of the boundary layer to simulate the local surface shape change as indicated in Figure 6.4. This was convenient as a first approximation. It is more correct to use the local value of velocity which means the multiplication factor in the leading-edge would be almost twice that used in the present analysis. This would lead to a further magnification in the trends noted toward the leading-edge. This velocity term is already saved in the RIM data base and the changes required to include this term are underway.

The data for the current problem under consideration were obtained at a very low Mach number and Reynolds number combination. The Whitfield program was developed to be valid into the transonic regime. Some modification to the code might be required at the low end of the spectrum to improve its applicability. Currently the Whitfield

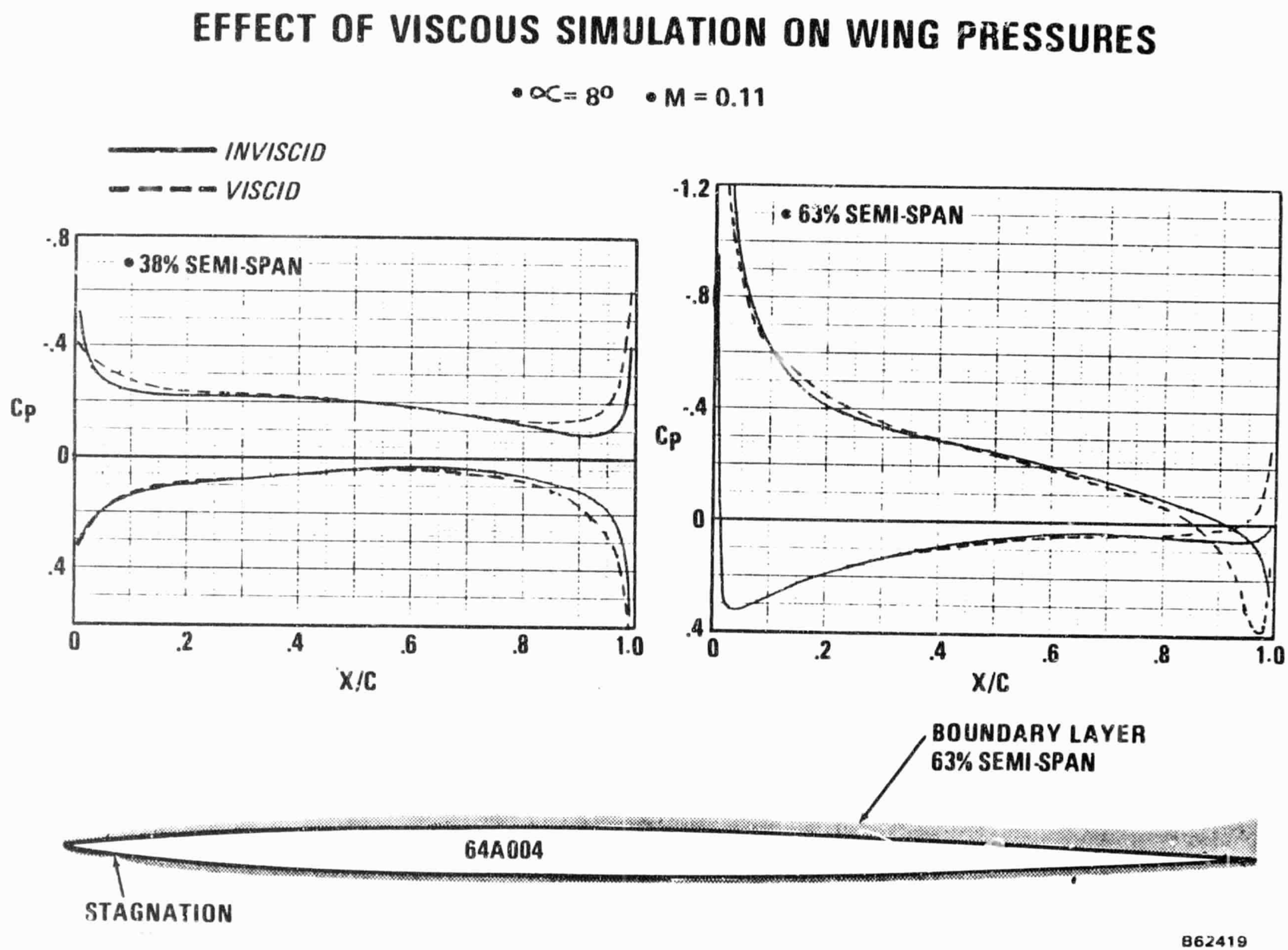


Figure 6-7 Effect of Viscous Correction on Chordwise Pressure Distribution

procedure estimates an initial value of boundary layer momentum thickness and shape factor based on Reynolds Number and Mach Number. Then the procedure uses a predictor/corrector scheme to march along the surface at a constant step size calculating the boundary layer properties. At some point downstream, the solution should be relatively independent of the initial conditions. The starting approximations and step size are being investigated for the present case.

Due to the previously discussed problems, a complete viscous solution involving several iterations was not achieved during the present study. Solutions to these problems are currently being investigated. The proper equivalent body to model viscous effects will evolve as the method matures.

7. CONCLUSIONS AND RECOMMENDATIONS

The development of a V/STOL analysis method has been initiated, and progress has been made in the development of three modules. Significant conclusions from this study are discussed below:

- The lift curve slopes computed by the PAN AIR code are generally in good agreement with experimental data. This is evidenced by the excellent agreement obtained in test-to-theory comparisions of lift curves for the Ames/Vought model discussed in this report and other configurations, such as the F-16XL analyzed by General Dynamics. However, the lift curves predicted for the Ames/GD model match the slope but not the level of the experimental data.

- The pressure coefficients predicted by the PAN AIR code are in good agreement with experimental data except in local regions where viscous effects are known to significantly influence the flowfield.

- The Preprocessor Module was found to be very useful in developing a panelling arrangement for the Ames/Vought VATOL model. Significant reductions in manual data manipulation were achieved.

- Effective use of the Preprocessor Module requires extensive use of graphics capability to develop and display the panelling schemes for complex regions of a configuration and to rapidly detect panelling errors. This need was demonstrated in the panelling arrangement developed for the Ames/Vought VATOL model. The graphics package developed by General Dynamics cannot be transferred to NASA until the package is modified to operate with the NASA software. This was not accomplished since extensive graphics capability and more sophisticated hardware is already available at NASA Ames.

- The Engine Simulation Module, which generates plume characteristics, predicted the correct trends in surface pressure changes with varying levels of nozzle pressure ratio. Boundary layer simulation on the nozzle afterbody is expected to further improve the predictions.

- The Engine Simulation Module can be used to simulate exhaust jets on configurations with non-complex nozzle arrangements. However, it cannot be used for complex configurations like the Ames/GD V/STOL fighter model.

- The Viscous Simulation Module can be used to simulate viscous effects on lifting surfaces. The communication link between PAN AIR and boundary layer calculation procedures has been established. The viscous effects computed for the Ames/Vought VATOL fighter model did not show the correct trends, which indicates problems either in computing boundary layer characteristics or in the definition of the equivalent body.

The progress made on the V/STOL Aerodynamic Prediction Technique is very encouraging and this work should be continued. Recommendations for future work are outlined below and are designed to supplement planned experimental V/STOL programs at NASA Ames.

- Refine the Preprocessor Module to (1) read and modify existing PAN AIR files, (2) automate the procedures for redefining networks, i.e., combining several networks into one network and/or separating one network into several, and (3) initiate the development of an automated capability to allow multicomponent intersections and corner-point matching at abutting networks.

- Refine the Engine Simulation Module to (1) allow more user interface during plume generation, i.e., panel density, plume size, center-line location, etc., (2) incorporate constraints into the plume generation module that will force abutments between the plume and specified points on the configuration or wakes, (3) develop an iterative procedure to account for the aircraft flowfield influence on the position and shape of the plume, (4) initiate the development of a capability for the analysis of models with jets at relatively large injection angles to the freestream, and (5) account for inlet flowfields.

- Continue the development of the Viscous Module to (1) define a satisfactory equivalent body model to simulate boundary layer buildup and trailing-edge separation effects on lifting surfaces, (2) define an equivalent body to simulate viscous effects on podlike surfaces, and (3) develop empirical corrections for vortical flows.

REFERENCES

1. Whitten, P. D.: Semi-Empirical Powered-Lift Methodology Presentation to AFFDL. General Dynamics Report No. ARM 139, April 1974.

2. Maskell, E. C.; and Spence, D. A.: A Theory of the Jet Flap in Three Dimensions, Proc. of the Royal Society of London. Series A, Vol. 251, 1959, pp. 407-425.

3. Whitten, P. D.; and Howell, G. A.: Investigations of the VEO-Wing Concept in an Air-To-Ground Role. AFFDL-TR-79-3031, March 1979.

4. Hebert, J.; Levinsky, E. S.; Ramsey, J. C.; Lauderman, E. C.; Altman, H. G.; and Barbee, L. G.: STOL Tactical Aircraft Investigation, Vol. II, Design Compendium. Convair Aerospace Division of General Dynamics, AFFDL TR 73-21, Vol. II, May 1973. (AD 766 941).

5. Kerney, K. P.: A Theory of the High-Aspect-Ratio Jet Flap. AIAA Journal, Vol. 9, No. 3, March 1971, pp. 432-435.

6. Roe, M. H.; Renselaer, D. J.; Quam, R. A.; et al. STOL Tactical Aircraft Investigation - Externally Blown Flap, Vol. II, Design Compendium. Los Angeles Aircraft Division, Rockwell International.

7. Spence, D. A.: The Lift Coefficient of a Thin, Jet-Flapped Wing. Proc. of the Royal Society of London, Series A, Vol. 238, No. 1212, Dec. 1956, pp. 46-68.

8. Spence, D. A.: The Lift on a Thin Aerofoil with a Jet-Augmented Flap. The Aeronautical Quarterly, Vol. IX, Pt 3, Aug. 1958, pp. 287-299.

9. Runciman, W. J.; Letsinger, G. R.; Ray, B. F.; and May, F. W.: STOL Tactical Aircraft Investigation, Vol II, Part I, Aerodynamic Technology: Design Compendium, Vectored Thrust/Mechanical Flaps. The Boeing Company, AFFDL TR 73-19, Vol. II, Part I, May 1973. (AD 766 639).

10. Wooler, P. T.; Kao, H. C.; Schwendemann, M. F.; Wasson, H. R.; and Ziegler, H.: V/STOL Aerodynamic Prediction Methods Investigation. Air Force Flight Dynamics Laboratory, Technical Report AFFDL TR 72-26, Jan. 1972, Vol. I, Theoretical Development of Prediction Methods (AD 752 557): Vol. II, Application of Prediction Methods (AD 752 558); Vol. III, Manual for Computer Programs (AD 752 559); Vol. IV, Literature Survey (AD 752 563).

11. Goldhammer, M. I.; Lopez, M. L.; and Shen: Methods for Predicting the Aerodynamic and Stability and Control Characteristics of STOL Aircraft, Volume 1, Basic Theoretical Methods, AFFDL-TR-73-146, December 1973.

12. Goldhammer, M. I.; and Wasson, N. F.: Methods for Predicting the Aerodynamic and Stability Characteristics of STOL Aircraft, Volume II, STOL Aerodynamic Methods Computer Program. AFFDL-TR-73-146, December 1973.

13. Goldhammer, M. I.; and Lopez, M. L.: Methods for Predicting the Aerodynamic and Stability Characteristics of STOL Aircraft, Volume III, Engineering Methods. AFFDL-TR-73-146, December 1973.

14. Beatty, T. D.; and Kress, S. S.: Prediction Method for Propulsive Induced Forces and Moments of V/STOL Aircraft in Transition/STOL Flight. Report Number NADC-77119-30, 31 July 1979.

15. Magnus, Alfred, E.: and Epton, Michael A.: PAN AIR - A Computer Program for Predicting Subsonic or Supersonic Linear Potential Flows About Arbitrary Configurations Using a Higher Order Panel Method, Volume I - Theory Document. NASA Contractor Report 3251.

16. Sidwell, Kenneth W.; Baruah, Pranab K.; and Bussoletti, John E.: PAN AIR - A Computer Program for Predicting Subsonic or Supersonic Linear Potential Flows About Arbitrary Configurations Using a Higher Order Panel Method, Volume II - User's Manual. NASA Contractor Report 3252.

17. Medan, Richard T.; et. al: PAN AIR - A Computer Program for Predicting Subsonic or Supersonic Linear Potential Flows About Arbitrary Configurations Using a Higher Panel Method, Volume III - Case Manual. NASA Contractor Report 3254.

18. Baruah, Pranab K.; et al: PAN AIR - A Computer Program for Predicting Subsonic or Supersonic Linear Potential Flows About Arbitrary Configurations Using a Higher Order Panel Method, Volume IV - Maintenance Document. NASA Contractor Report 3254.

19. Moran, J.; Tinoco, E. N.; and Johnson, F. T.: User's Manual, Subsonic/Supersonic Advanced Panel Pilot Code. NASA CR-152047, February 1978.

20. Howell, G. A.; and Bhateley, I. C.: Theoretical Evaluation of a V/STOL Fighter Model Utilizing the PAN AIR Code. NASA CR-166395, September 1982.

21. Harris, M. J.; and Falarski, M. D.: Static Calibration of a Two-Dimensional Wedge Nozzle With Thrust Vectoring and Spanwise Blowing. NASA Technical Memorandum 81161, April 1980.

22. Falarski, M. D.; Whitten, P. D.; and Harris, M. J.: Aerodynamic Charcteristics of a Large-Scale Model of a Highely Maneuverable Supersonic V/STOL Fighter: STOL Configuration. AIAA Paper No. 80-0234, January 1980.

23. Howell, G. A.; Crosthwait, E. L.; and Witte, M. C.: Evaluation of Pressure and Thermal Data From a Wind Tunnel Test of a Large-Scale Powered, STOL Fighter Model. NASA CR-166170, June 1981.

24. Falarski, M.; Dudley, M.; and Howell, G.: Analysis of Data From a Wind Tunnel Investigation of a Highly Maneuverable Supersonic V/STOL Fighter: STOL Configuration. AIAA Paper No. 81-2620, December 1981.

25. Howell, G. A.: Test Results of Chordwise and Spanwise Blowing for Low-Speed Lift Augmentation. 13th Congress of the International Council of the Aeronautical Sciences, Paper No. ICAS-82-5.7.1, August 1982.

26. Stoll, F.; and Minter, E.: Large-Scale Wind Tunnel Tests of a Sting-Supported V/STOL Fighter Model at High Angles of Attack. AIAA Paper No. 81-2621, December 1981.

27. Minter, E. A.; and Yates, R. W.: Wind Tunnel Test of a .4 Scale Fighter Model at High Angles of Attack - Analysis of Pressure Data. NASA CR-160198, July 1981.

28. Halsey, N. Douglas; and Hess, John L.: A Geometry Package for Generation of Input Data For a Three-Dimensional Potential Flow Program. NASA Contractor Report CR-2962, March 1978.

29. Hedgley, Jr., David R.: A General Solution to the Hidden-Line Problem. NASA Reference Publication 1085, 1982.

30. Putnam, L.; and Mace, J.: A Survey of Aft Body Flow Prediction Methods, AIAA Paper No. 81-1694, August 1981.

31. Cline, Michael C.: VNAP 2: A Computer Program for Computation of Two-Dimensional, Time-Dependent, Compressible, Turbulent Flow. Los Alamos National Laboratory Rport No. LA-8872, August 1981.

32. Wilmoth, Richard G.: RAXJET: A Computer Program for Predicting Transonic, Axisymmetric Flow Over Nozzle Aftebodies With Supersonic Jet Exhausts. NASA Technical Memorandum 83235, February 1982.

33. South, Jerry C., Jr,; and Jameson, Antony: Relaxation Solutions for Inviscid Axisymmetric Transonic Flow Over Blunt or Pointed Bodies. AIAA Computational Fluid Dynamics Conference, July 1973, pp. 8-17.

34. Keller, James D.; and South, Jerry C., Jr.: RAXBOD: A Fortran Program for Inviscid Transonic Flow Over Axisymmetric Bodies. NASA TM X-72831, 1976.

35. Dash, Sanford M.; Pergament, Harold S.; and Thorpe, Roger D.: Computational Models for the Viscous/Inviscid Analysis of Jet Aircraft Exhaust Plumes. NASA CR-3298, 1980.

36. Abbett, Michael: Mach Disk in Underexpanded Exhaust Plumes. AIAA J., Vol. 9, No. 3, March 1971, pp. 512-514.

37. Reshotko, Eli; and Tucker, Maurice: Approximate Calculation of the Compressible Turbulent Boundary Layer With Head Transfer and Arbitrary Pressure Gradient. NASA TN 4154, 1957.

38. Presz, Walter M., Jr.; King, Ronald W.; and Buteau, John D.: An Improved Anaytical Model of the Separation Region on Boattail Nozzles at Subsonic Speeds. NASA CR-3028, 1978.

39. User's Manual for the External Drag and Internal Nozzle Performance Deck (Deck XI) - Supersonic Flow Analysis (Applicable to Deck VI): PWA-3465, Suppl. F, Pt. I (Contract No. AF33(615)-3128), Pratt and Whitney Aircraft, Sept. 1, 1968.

40. Presz, Walter Michael, J.: Turbulent Boundary Layer Separation of Axisymmetric Afterbodies. Ph. D. Thesis, Univ. of Connecticut, 1974.

41. Peters, C. E.; and Phares, W. J.: An Integral Turbulent Kinetic Energy Analysis of Free Shear Flows. Free Turbulent Shear Flows. Volume I - Conference Proceedings, NASA SP-321, 1973, pp. 577-624.

42. Dash, Sanford M.; and Pergament, Harold S.: A Computational Model for the Prediction of Jet Entrainment in the Vicinity of Nozzle Boattails (The BOAT Code). NASA CR-3075, 1978.

43. Berrier, Bobby L.; and Re, Richard Jr.: Investigation of Covergent - Divergent Nozzles Applicable to Reduced - Power Supersonic Cruise Aircraft. NASA Technical Paper 1766, December 1980.

44. Whitfield, David L.: Integral Solution of Compressible Turbulent Boundary Layers Improved Velocity Profiles. AEDC-TR-78-42, December 1978.

APPENDIX A

USER'S GUIDE TO THE PREPROCESSOR MODULE

The appendix gives the user instructions required to run the Preprocessor Module, sample inputs, and results. Listings of the source codes developed specifically for this module are included in section A.2. This module was developed to be operational on any of the NASA Ames' VAX 11/780's.

A.1 Preprocessor Module Command File (PANEL.COM)

The execution of the Preprocessor module is controlled by a command file (designated PANEL.COM). Every function performed by the Preprocessor can be executed by keying in the proper response to the prompts issued by this command file. If it is desired to operate one of the Preprocessor programs without entering the command file, any individual function may be performed by addressing the required program directly after making the proper assignment commands. Execution of the Preprocessor Module is initiated as follows:

1. Press "@ PANEL" (RETURN)

2. The VAX will respond with a listing of the "Terminal Type Indentifier Codes" and ask the user to "Enter Terminal Type". The user enters the integer code corresponding to the terminal being used. For example, for a VT100 terminal, the response is:

   1 (Return)

3. The VAX responds with a "WHAT NOW?" prompt. The available commands are:

   (a) HCPY - Run RASM on plot file
   (b) PGEN - Create a PAN AIR input file
   (c) CFIL - Create preliminary geometry file
   (d) EXEC - Execute panel generation procedure

(e) INPT - Generate input file to panel generation procedure

(f) MODD - Modify geometry files

(g) PLOT - Plot geometry files

(h) LOUT - Logout

(i) HELP - List available commands

The user now enters one of the above four letter commands to perform the desired task. For example, to plot a geometry file, the following command is entered:

PLOT (RETURN)

Execution now shifts to the plot program. The programs called by each of these commands are shown in Figure A-1. Each of these programs may call other programs. After execution of the plot program, control is returned to the command file.

The code contained in PANEL.COM is included in Section A.2.1. Each of the available commands which result in a transfer to another program are further explained in the following sections along with a utilization for that program. As previously stated, each of these programs can be directed from the command file or executed separately if the user desires.

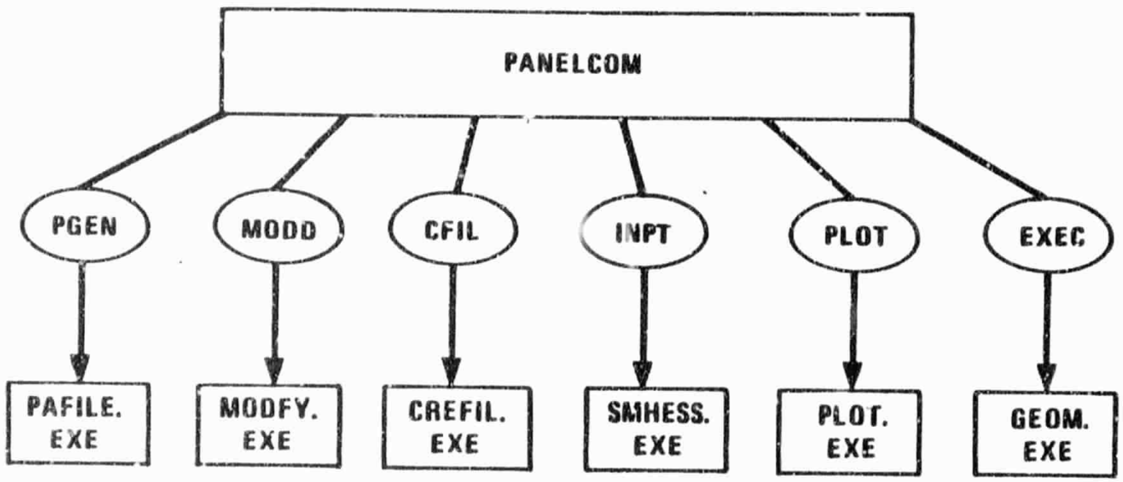
The commands which do not result in a transfer to another program are explained below.

1. HCPY: During execution of the graphics program which is initiated by the PLOT command, the coordinates of any view may be saved in a file. After transfer of control back to the command file, a plot of these views may be obtained on the VERSATEC printer by executing the HCPY command.

2. HELP: The available commands for execution from the command file are listed on the terminal screen by this command.

3. LOUT: This command transfers control out of the command file at the end of any session.

PREPROCESSOR MODULE COMMAND FILE



- PGEN: ADDS JCL AND PANAIR INPUT PROBLEM TO FILE CONTAINING PANEL DEFINITION
- MODD: TO EDIT AN EXISTING GEOMETRY FILE
- CFIL: GENERATES A PRELIMINARY GEOMETRY FILE BY PROMPTING USER FOR COORDINATE DEFINITION
- INPT: ADDS USER SPECIFIED REPANELLING OPTIONS TO PRELIMINARY GEOMETRY FILE TO CREATE INPUT FILE TO GEOMETRY PACKAGE
- PLOT: PLOTS USER SPECIFIED FILE
- EXEC: INITIATES EXECUTION OF GEOMETRY PACKAGE TO PANEL CONFIGURATION AS SPECIFIED BY USER

962336

Figure A.1 Programs Directed From Preprocessor Module Command File

A.1.1 Program to Create a Preliminary Geometry File (Command: CFIL)

The purpose of this program is to help the user generate a preliminary geometry file to be used in the panel generation program. It is assumed that the user has the X, Y, Z coordinate definition of each component and wants to record them in a VAX file to initiate development of a detailed paneling scheme for a PAN AIR problem. The coordinate definition may be very sparse as the geometry package will later augment and redistribute these points as desired by the user. This program will prompt the user for the number of points per column, the number of columns, and then a point by point data entry. The code for this program is presented in Section A.2.4. A program utilization is not required becuase the user is prompted for each input as required; however, a brief discussion of what the user will be prompted for follows.

The user is first asked to enter one of the the following codes to indicate whether additions are to be made to an existing file or entries are to be made to a new file. This prompt is as follows:

"Enter the code corresponding to the method that you desire. Enter the negative of the code to continue a previous geometry definition."

0 - Abort
1 - Point by point data entry
2 - Interactive configuration development

The user is currently limited to entering either a 0 or 1 because the interactive development option is not available.

For a point by point data entry, the user is next asked to enter the name of the preliminary geometry file name. This may be either the name of an existing file to which additions are to be made or the name of a new file to be generated. The next prompt is for a title to be placed on the first line of the file.

The user is now prompted for a network title, the number of points per column, and the number of columns. The program then prompts the user to enter each X, Y, Z point (format as follows: XXX.XXX, YYY.YYY, ZZZ.ZZZ) until the entire network is defined. As many decimal places as desired may be used; however, each X, Y, Z must be entered

on a single line separated by commas. After completion, which is indicated by entering "QUIT", the data is stored in a VAX file under the name which was previously indicated.

This program may be executed without entering the Preprocessor Command File (PANEL.COM) by entering the VAX command "RUN WRTCON." A preliminary geometry file may also be generated in the VAX editor mode by observing the following formats when creating the file.

Line 1: Title (any desired title in A format)
Line 2: M N NTITLE (title card for first network)
Line 3: X Y Z (begin coordinate definition of first network, one point per line)
Note: Repeat beginning at line 2 for each network.

where: M = number of points per column
N = number of columns
NTITLE = network title
X,Y,Z = coordinates of point

Each network title card is entered in the format (2I4, A30) and each X,Y,Z coordinate is entered in the format (3F20.5). The file is complete when all networks are defined.

A.1.2 Program To Create An Input File To The Panel Generation Procedure (Command: INPT)

After the preliminary geometry file has been created as described in Section A.1.1, the input options for the Halsey/Hess procedure need to be combined with the geometry file to create an input file to the panel generation program (GEOM.EXE). These input options basically indicate the spacing options and number of panels desired for the final geometry defintion for the PAN AIR problem. The program (SMHESS.EXE) written to perform this task is written in a conversational mode. The program defines each desired input then prompts the user for a value. A definition of the input parameters required for the Halsey/Hess geometry package is provided in Table A-1. Execution of this program will basically flow as follows:

1. User will be prompted for name of preliminary geometry file (13 characters max.). Input name of file which was created as described in Section A.1.1.

"Enter Preliminary Geometry File Name"

2. Default values for input options to Halsey/Hess procedure will be reviewed on terminal. These values (refer to Table A-1) are as follows.

"Default Values for Input to the Panel Generation Procedure Are:"

NOPT = 1
IMODE = 0
NPTS = 1
NSEG = 1
NB = 0
ITR = 0
NW = 0
INTS = 0
NS = 0
NALG2 = 0
NALGB = 0
NALG3 = o
NALGW = 0
NALGS = 0

3. "Do you wish to review all program options and variables and/or only those selected by the user." The user now has the option to review all of the program options and variables (including the above default options) or only those specified by the user. If user selects "NO", only those variables specified by user will be changed from the default value. If user selects "YES", then each variable and/or option will be defined and the user will be asked to enter a value.

   A. "YES" is entered by user. Each variable presented in Table A-1 is defined and the user is prompted for a value. If no input is required for a particular variable due to a previously selected option, then that variable is

not reviewed. After all options are selected for a particular network (or component) then the options and their values are presented on the terminal. After all values are entered, the user is offered the opportunity to change any value by specifying the variable and component number as in (B) below. Enter "Q" if finished.

B. "NO" is entered by user. With this answer all default values remain unchanged unless a variable including the component number is specified in the following format: VAR (I) = VAL where VAR is one of the input options or variables defined in Table A-1, I is a component number (if required), and VAL is the value desired for VAR. A printed listing of the variables in Table A-1 is available if requested. Enter "Q" when finished.

4. "Enter Input File Name (13 characters)". After all variables have been entered, the user is prompted for the name of the file to store the input variables for the Halsey/Hess geometry procedure.

5. "Enter Output File Name (13 characters)" Enter here the name of the VAX file where the results from the Halsey/Hess geometry procedure will be stored. This file will contain the network definition to be used with the PAN AIR procedure and is the primary output file.

6. "Enter Batch Output File Name (13 characters)" Enter here the file name where the input, results, and intermediate results from execution of the Halsey/Hess geometry procedure will be stored. This file is primarily suited to be batched to a printer for reviewing the input and results from the Halsey/Hess procedure. This will not be used in conjunction with PAN AIR and may be purged if it is not desired.

After the above output files for the geometry package results are identified, control is shifted to a command file (ASSGN.COM) which makes the logical unit assignment for execution of the Halsey/Hess geometry procedure. Control is then shifted back to the Preprocessor Module Command File (PANEL.COM). The execution of the geometry procedure is then initiated with the "EXEC" command as described in the following section.

The input file for the Halsey/Hess procedure may be generated in two ways without entering the Preprocessor Module Command File (PANEL.COM). The execution of the previously described program may be initiated by entering the VAX instruction: "RUN SMHESS". All of the preceeding discussion on input options and variables would then apply.

The input file generated by this program (SMHESS.EXE) could also be generated in the VAX editor mode. The file generated in this manner should conform to the formats described in Table A-2.

The recommended ordering of input coordinates is shown in Figure A-2. All points on the first N-line (including wake points which are entered last for each N-line on a lifting component) are input consecutively, followed by all points on the second N-line, and continued in this manner until all N-lines for that component are entered. The first N-line may be at either extreme of the component, but the choice determines the order of input of points along the N-lines. The order must be such that the negative of the cross product of the vector from one point on an N-line to the next point on the N-line with the vector from a point on an N-line to the corresponding point on the next N-line results in a vector which is directed to the exterior of the component. On a wing, this requirement is satisfied by ordering the N-lines from tip to root while ordering points on the N-lines from the trailing-edge along the lower surface to the leading-edge and back along the upper surface to the trailing-edge. On a fuselage, this requirement is satisfied by ordering the N-lines from front to back and the points on each N-line increasing counterclockwise (looking aft).

The code developed for the program (SMHESS) to create an input file for the Halsey/Hess Procedure and an example input file are presented in Section A.2.2.

A.1.3 Execution of the Panel Generation Procedure (Command: EXEC)

Once the input file described in Section A.1.2 has been created, the geometry procedure may be run. If control has been transferred back to the Preprocessor Command File, this may be accomplished by answering the prompt "WHAT NOW" with the command "EXEC". The "INPT" command which initiates the running of the SMHESS.EXE program must be used before the "EXEC" command is used. The reason being that the assignment of the logical units are made during execution of the SMHESS.EXE program.

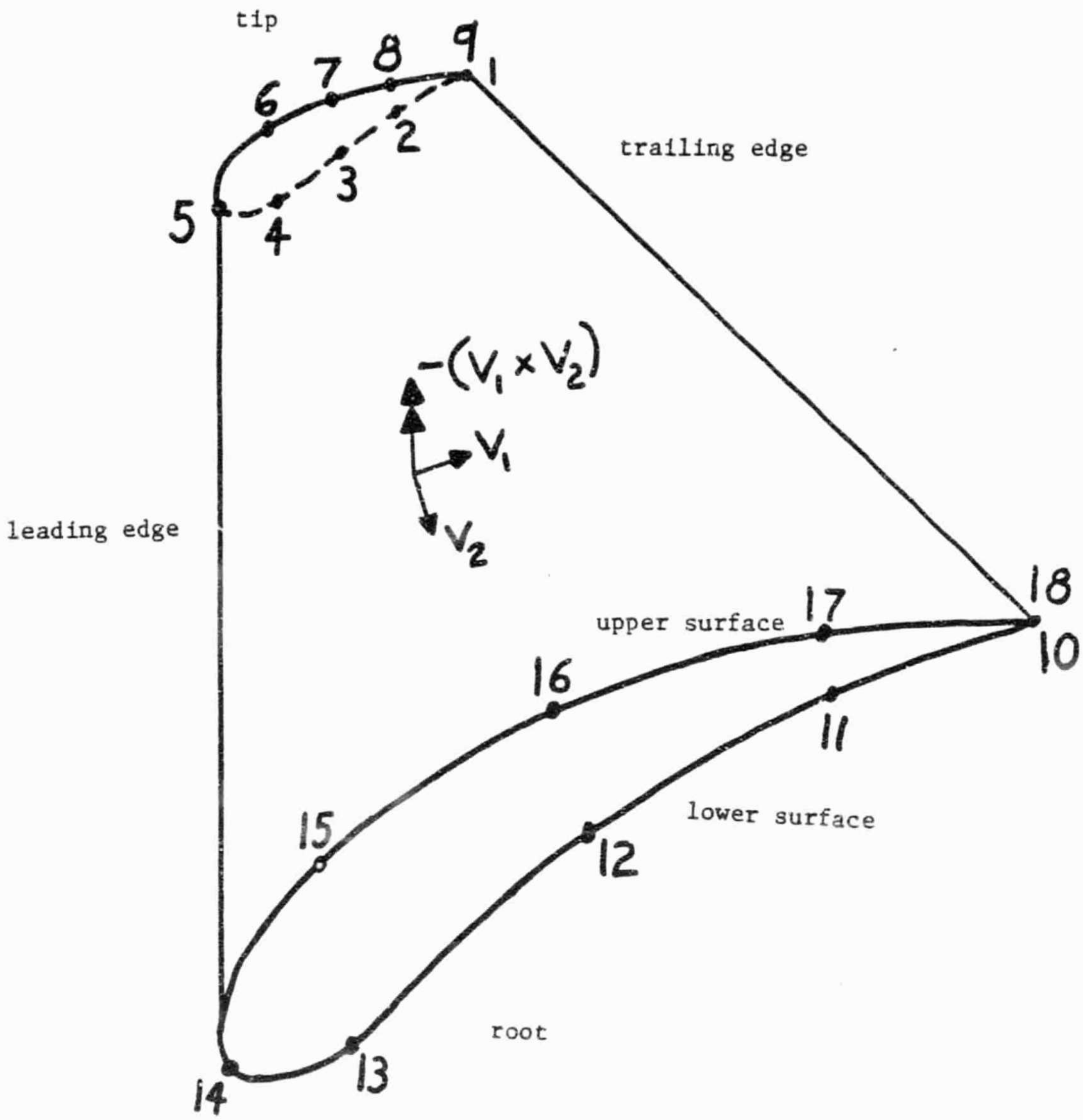


Figure A-2 Recommended Ordering of Points to Define a Wing

If the user desires to run the geometry procedure without entering the Preprocessor Command File (PANEL.COM), then the following steps must be taken.

1. Generate input file as described in Section A.1.2.

2. Make assignment of logical units on VAX as follows:

   (i) ASSIGN FILE.DAT FOR005 (FILE.DAT is the input file created in (1) above)

   (ii) ASSIGN BATCH.PRT FOR006 (BATCH.PRT is the output file from the Halsey/Hess Procedure which is suitable to be batched to a printer)

   (iii) ASSIGN OUT.DAT FOR004 (OUT.DAT is the output file from the Halsey/Hess Procedure which contains the panel coordinate data to be used for PAN AIR input.

   (iv) ASSIGN SYS$OUTPUT FOR008

The names of the above files may be selected to suit the user.

3. The geometry package may now be run using the VAX command "RUN.GEOM".

A.1.4 Graphics Package (Command: PLOT)

From the Preprocessor Command File (PANEL.COM) the graphics package may be utilized using the "PLOT" command. A preliminary geometry file (Section A.1.1), an output file from the Halsey/Hess Procedure (Section A.1.3), or a PAN AIR input file may be plotted. The "PLOT" command results in the transfer of control to the program THRDPL (Three Dimensional Plots).

The user encounters the following prompts as illustrated in Figure A.3 after using the "PLOT" command.

(1) "Enter File Name". Enter file to be plotted.

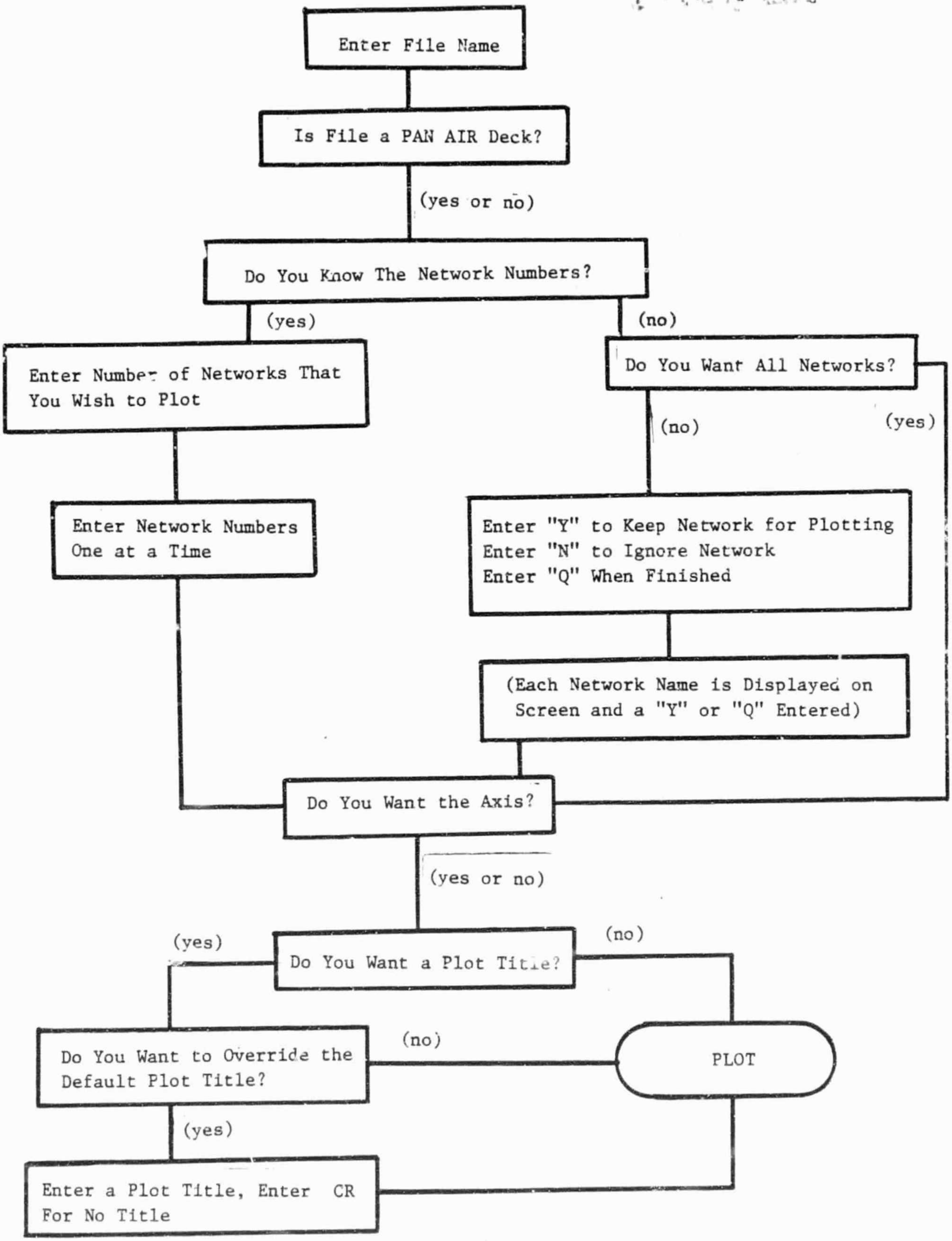


Figure A-3 **Flow Chart for Graphics Package**

(2) "Is file a PAN AIR Deck": Enter "Yes" or"No". If "Yes" or "No" then the following prompts are encountered:

(3) "Do You Know the Network Numbers?" If "Yes" to (3) then

(3i) "Enter Number of Networks that You Wish to Plot". Enter number.

(3ii) "Enter Network Numbers One At a time". Enter network networks.

If "NO" was the answer to (3), then the following prompts are encountered.

(3a) "Do You Want All Networks?" Enter "Yes" or "No".

If answer is "Yes" then question (4) is next. If answer is "No" then each network title is displayed and a "Yes" or "No" answer is expected for each.

(4) "Do You Want the Axis?: Enter "Yes or No".

(5) "Do You Want a Plot Title?": Enter "Yes" or "No".

If answer is "Yes" then the following prompts are encountered:

(5i) "Do You Want to Override the Default Plot Title?" (Default is the file title card) If "Yes" to (5i) then:

(5ii) "Enter a Plot Title, Enter Return for No Title": Enter a Title.

(6) A frontal view of the configuration is now plotted on the terminal. The interactive plot manipulation commands shown in Table A-3 are now available to move the configuration to the desired viewing perspective. The initial pivot point for rotating the configuration is set at (0,0,0) so it is desirable to use the "PIVT" command from Table A-3 to a point near the middle of the configuration.

(7) Use "STOP" command to terminate the plotting sequence.

A.1.5 Program to Modify Selected Points of the Detailed Panelling Scheme (Command: MODD)

During plotting and checking of the detailed panelling scheme generated for the PAN AIR problem, sometimes it becomes necessary to edit or change selected coordinates. Editing is allowed during the plotting sequence (Section A.1.4) by using the interactive plot manipulation command "CHNG" from Table A-3. Sometimes it is desirable to edit a selected few points while in the Preprocessor Command Module without entering the plot sequence. This capability is provided with the command "MODD" which can be executed from the Preprocessor Command File (PANEL.COM). After exercising the "MODD" command, the user is presented the following prompts:

- "Enter File Name to be Modified": User enters the desired file name.

- "Enter Point Number for Changes": The points are assumed to be numbered beginning with the first point in the first network as 1-1 (i.e., 1-1, 1-2, 1-3..., 2-1, 2-2, 2-3..., N-1, N-2, N-3....) where the first number is the network number and the second is the relative position of the point within that networks definition. During the plot seequence (Section A.5), these numbers may be displayed using the interactive manipulation command "NUMB".

- -Enter "Q" when the last point is changed.

The code developed for the "MODD" command is presented in Section A.2.3. The editing process may also be accomplished in the VAX editor mode without entering the Preprocessor Command Module.

A.1.6 Program to Create a PAN AIR Input Problem (Command: PGEN)

This command simply asks the user for the geometry file created by the Halsey/Hess procedure and adds the JCL and input records required for a PAN AIR problem. It should be noted that the user will be required to specify the type of boundary conditions, abutment specifications, etc. by entering the VAX editor mode. This program simply adds the required JCL and makes the PAN AIR program options that are required for this methodology. The correct positions for the specification of boundary conditions, abutments, etc, are indicated.

Table A-1: DESCRIPTION OF INPUTS TO HALSEY/HESS GEOMETRY PROCEDURE (GEOM.EXE)

DESCRIPTION OF INPUTS

Variable		Description
NSEC	-	Total number of components
NPTS	-	Format control for X,Y,Z, input coordinate data.
		NPTS = 1, 1 point per record.
		NPTS = 1, 2 points per record (not available) Note: NPTS automatically set equal to 1 for this application.
NOPT	-	Execution flag.
		NOPT < 2, panel isolated components only.
		NOPT = 2, panel isolated components and calculate intersection curves.
		NOPT > 2, panel isolated components, calculate intersection curves, and repanel intersecting and intersected components.
NB(I)	-	Specified number of on-body points per N-line on component I.
NALGB(I)	-	Spacing algorithm for on-body points on N-lines for component I.
		NALGB = 0, input distribution, unaltered.
		NALGB = 1, input distribution, augmented.
		NALGB = 2, constant increments in arc length.
		NALGB = 3, cosine distribution.
		NALGB = 4, curvature-dependent distribution.
		NALGB = 5, user-specified distribution.

(Table A-1 Continued)

Variable		Description
NW(I)	-	Specified number of wake points per N-line on component I.
NALGW(I)	-	Spacing algorithm for wake points on N-lines. (Values have same significance as values of NALGB, except that NALGW = 3 and NALGW = 4 should not be used.)
NS(I)	-	Specified number of N-lines on component I.
NALGS(I)	-	Spacing algorithm for N-lines on component I. NALGS = 0, input distribution, unaltered. NALGS = 1, input distribution, augmented. NALGS = 2, constant increments. NALGS = 3, user-specified distribution.
INTS(I)	-	Index of component (if any) which component I intersects.
NALG2(I)	-	Repaneling for intersecting components if I is an intersecting component. NALG2 = 0, minimal repaneling. NALG2 = 1, full repaneling.
NALG3(I)	-	Repaneling flag for intersected components if I is an intersected component. NALG3 = 0, no repaneling. NALG3 = 1, (a) Nonlifting components. Full repaneling with an N-line passing through every point on intersection curve. (b) Lifting components. Full repaneling, but with gaps in the region of the intersection curve.

(Table A-1 Continued)

Variable		Description
		NALG3 = 2, (a) Nonlifting components. Full repaneling with an N-line passing through every second point on the intersection curve. (b) Lifting components. Full repaneling with additional nonlifting elements to fill the gaps around the intersection curve.
IXFLAG(I)	-	Extra strip flag for component (I). IXFLAG = 0, no extra strip. IXFLAG = 1, the first strip of the final coordinates is extra. EXFLAG = 3, the last strip of the final coordinates is extra.
IMODE(I)	-	Planar-section/arc-length model flag for component I. IMODE = 0, planar-section mode using default values of direction cosines of axes (1,0,0) for nonlifting components; (0,-1,0) for lifting components). IMODE = 1, planar-section mode using input values of direction cosines of axis. IMODE = 2, arc-length mode.
COSX(I)	-	X-direction cosine of the axis of component I.
COSY(I)	-	Y-direction cosine of the axis of component I.
COSZ(I)	-	Z-direction cosine of the axis of component I. NOTE: COSX(I), COSY(I), and COSZ(I) are required if IMODE(I) = 1.
ITR(I)	-	Component transformation flap. (The value indicates the number of transformations to be performed.)

(Table A-1 Continued)

Variable		Description
NSEG(I)	-	Number of segments into which M-lines are broken (Default = 1).
SSB(J,I)J=1,NB(I)	-	Specified arc lengths of on-body points on N-lines on component I. $o.o \leq SSB \geq 1.0$. NOTE: NB(I) total points per component only required if NALGB(I) = 5.
SSW(J,I),J=1,NW(I)	-	Specified arc lengths of wake points on N-lines on component I. $0.0 \leq SSW \geq 1.0$. <u>NOTE</u>: NW(I) total points per component only required if NALGW(I) = 5.
SSS(J,I),J=1,NS(I)	-	Specified distribution of N-lines on component I. $0.0 \leq SSS \geq 1.0$. <u>NOTE</u>: NS(I) total points per component only required if NALGS(I) = 3.
IEND(J,I),J=1,NSEG(I)	-	Point numbers of the ends of the segments on M-lines on component I. <u>NOTE</u>: NSEG(I) total (max. = 5) per component only required if NSEG(I) > 1.
ITR2	-	Type of transformation to be applied to component I. IRT2 = 1, scaling ITR2 = 3, translation ITR3 = 3, rotation
TR(1)	-	First transformation parameter: x multiplication factor if ITR2 = 1 x translation if ITR2 = 2 angle of rotation if ITR2 = 3 (Positive clockwise, looking down axis away from origin).

(Table A-1 Continued)

Variable		Description
TR(2)	-	Second transformation parameter: y multiplication factor if ITR2 = 1 y translation if ITR2 = 2 x direction cosine of axis of rotation if ITR 2 = 3
TR(3)	-	Third transformation parameter: z multiplication factor if ITR2 = 1 z translation if ITR2 = 2 y direction cosine of axis of rotation if ITR2 = 3
TR(4)	-	Fourth transformation parameter: Dummy if ITR2 = 1 or ITR2 = 2 z direction cosine of axis of rotation if ITR2 = 3 NOTE: The axis of rotation is assumed to pass through the origin. This input is required if ITR(I) ≠ 0 for any component I. A total of ITR(I) transformations are performed for each component I.
X Y Z	-	X,Y,Z coordinates of a point. (All points input N-line by N-line, component by component.)
STATUS	-	STATUS = 0 same N-line STATUS = 1 new N-line STATUS = 2 new component STATUS = 3 last point input (exception below) STATUS = 4 first point on wake STATUS = 5 point on wake and also last point input

(Table A-1 Continued)

Variable		Description
LABEL	-	LABEL = 0 nonlifting component LABEL = 1 lifting component
X Y Z		X,Y,Z coordinates of the next point input.
STATUS LABEL		Same meaning as above.
		NOTE: Repeat records to input more points. If NPTS = 1, input only 1 point per record (columns 1-32).
		NOTE: The X,Y,Z coordinate data will be read from the preliminary geometry file created as described in Section A.2. The status of each point will be added after each point in the format (3F 20.5, IX, 2I1).

TABLE A-2 FORMAT FOR THE HALSEY/HESS INPUT FILE

Line Number	Variables	Format
Line 1	Format for the Halsey/Hess Input File	(A60)
Line 2	NSEC, NPTS, NOPT	(IX, I2, IX, I1, IX, I1)
Line 3	NB(I), NALGB(I), NW(I), NALGW(I), NS(I), NALGS(I), INTS(I), NALG2(I), NALG3(I), IXFLAG(I), IMODE(I), ITR(I), NSEG(I), NTIT(I)	(IX, I2, IX, I1, IX, I2, IX, I1, IX, I3, IX, I1, IX, I2, IX, I1, IX, I1, IX, I1, 1X, I1, IX, I1, IX, I1, IX, A30)
Line 4-I	Line 3 is repeated until entered for all I components to be defined.	
*I+1	COSX(I), COSY(I), COSZ(I) *Not required unless IMODE(I) = 1	3F10.5
**I + 2	SSB (J,I) **Not required unless NALGB(I) = 5. Enter NB(I) values (one value per line)	F10.5
***I + 3	SSW (J,I) ***Not required unless NALGW(I) = 5. Enter NW(I) values (one value per line).	(F10.5)
****I + 4	SSS (J,I) ****Not required unless NALGS(I) = 3. Enter NS(I) values (one value per line).	F10.5)
*****I + 5	IEND(I) *****Not required unless NSEG(I) > 1. Enter NSEG(I) values (one value per line).	(I4)
******I + 6	ITR2 (J,I), TR1(J,I), TR2(J,I), TR3(J,I), TR4(J,I) ******Not required unless ITR(I) > 0. Enter ITR(I) lines of the variables.	(I1,4F10.5)

(TABLE A-2 Continued)

NOTE: All optional input marked with asterisks (lines I+1 thru I+6) are entered for each component. (i,e, lines I+1 thru I+6 are entered for component 1 as required, then for component 2 as required, etc.) All of the above inputs are entered before going on to the coordinate data.

I + 7	X, Y, Z, STATUS, LABEL	(3F20.5, 1X, 2I1)

NOTE: Enter all coordinate starting with the first point of component 1 until all components are entered (1 point per line).

Table A.3 Interactive Plot Manipulation Commands

THREE-D PLOT MANIPULATION COMMANDS

HELP	- Produce a list of 3-D plot manipulation commands
RSET	- Reset plot to first condition (before any user manipulation)
SROT [VAL]	- Set the incremental rotation angle in degrees (Default is 15 degrees)
STRN [VAL]	- Set the incremental translation in screen widths (Default is 0.1)
SSCL [VAL]	- Set the incremental scale factor (NEWSCAL=SCAL*SCINC)
HALF	- Plot only the half model (Default)
FULL	- Plot a full model (reflect the half model through the centerline, Y=0, plane)
PRSP	- Display model with perspective projection (Default)
ORTH	- Display model with orthographic projection
OBSV [VAL]	- Set the position of the observer (Default x = -500)
PROJ [VAL]	- Set the position of the projection plane (Default X= -200)
PIVT	- Define the pivot point about which the rotations occur (Default X=0,Y=0,Z=0)
RLRT [VAL]	- Roll configuration to pilot's right
RLLT [VAL]	- Roll configuration to pilot's left
PTUP [VAL]	- Pitch the configuration in the positive direction
PTDN [VAL]	- Pitch the configuration in the negative direction
YWRT [VAL]	- Yaw the configuration to the pilot's right
YWLT [VAL]	- Yaw the configuration to the pilot's left
SCUP [VAL]	- Enlarge the picture
SCDN [VAL]	- Reduce the picture size
TRUP [VAL]	- Translate the picture towards the top of the screen
TRDN [VAL]	- Translate the picture towards the bottom of the screen
TRRT [VAL]	- Translate the picture towards the right side of the screen
TRLT [VAL]	- Translate the picture towards the left side of the screen
HIDN	- Remove hidden lines
SHOW	- Display all lines (default)
PLAN	- Generate a planform view
PROF	- Generate a profile view
HCPY	- Replot display and generate a hardcopy
NUMB	- Show point identification numbers on the graphics display
CHNG	- Enter a change in position for one or more points
STOP	- End plotting sequence

A.2 FORTRAN Codes Developed for Preprocessor Module

This appendix contains codes developed specifically for the Preprocessor Module. The methods included in the Preprocessor Module were described in Section 4. A utilization on how to use this module was included in section A.1. The codes written in the VAX Fortran were separated into this section for ease of referral.

A.2.1 Code for PANEL.COM

The following pages contain the code developed for the PANEL.COM procedure.

ORIGINAL PAGE IS
OF POOR QUALITY

```
!
!     PANEL.COM
!
!          SUPERVISE THE EXECUTION OF THE INTERACTIVE
!       AERODYNAMIC PANELING PROCEDEURE
!

!         DEFINE ROUTINES AS FOREIGN DCL COMMANDS
!
$CREFIL:==$DRCO:[BHATELEY.PANAIR]CREFIL.EXE
$SMHESS:==$DRCO:[BHATELEY.PANAIR]SMHESS.EXE
$PLTT:==$DRCO:[BHATELEY.PANAIR]PLOT.EXE
$MODFY:==$DRCO:[BHATELEY.PANAIR]MODFY.EXE
$GEOM:==$DRCO:[BHATELEY.PANAIR]GEOM.EXE
!
!         DETERMINE WHICH TERMINAL IS BEING USED
!
$LAB10:
!
$WRITE SYS$OUTPUT " TERMINAL TYPE IDENTIFIER CODES"
$WRITE SYS$OUTPUT " "
$WRITE SYS$OUTPUT " 1 - VT100"
$WRITE SYS$OUTPUT " 2 - TEKTRONIX 4025"
$WRITE SYS$OUTPUT " 3 - TEKTRONIX 4014"
$WRITE SYS$OUTPUT " 5 - TEKTRONIX 4010"
$WRITE SYS$OUTPUT " 6 - TEKTRONIX 4027"
$WRITE SYS$OUTPUT " "
$INQUIRE NTERM "ENTER TERMINAL TYPE"
$ASSIGN SYS$INPUT FOR005
$ASSIGN SYS$OUTPUT FOR006
$LAB50:
!
!                    SELECT ROUTINE TO EXECUTE
!
$INQUIRE EXEC "WHAT NOW?"
$IF EXEC.NES."CFIL" .AND. EXEC.NES."PLOT"-
.AND. EXEC.NES."INPT" .AND. EXEC.NES."EXEC"-
.AND. EXEC.NES."MODD".AND. EXEC.NES."LOUT"-
.AND. EXEC .NES."HCPY".AND. EXEC .NES."PGEN"-
.AND. EXEC.NES."HELP" THEN GOTO LAB50
$GOTO 'EXEC'
$GOTO LAB50
!
$HELP:
!
!           LIST AVAILABLE COMMANDS
!
$WRITE SYS$OUTPUT "  "
$WRITE SYS$OUTPUT " HCPY - RUN RASM ON PLOT FILE"
$WRITE SYS$OUTPUT " PGEN - CREATE A PANAIR INPUT FILE"
$WRITE SYS$OUTPUT " CFIL - CREATE PRELIMINARY GEOMETRY FILE"
$WRITE SYS$OUTPUT " EXEC - EXECUTE PANEL GENERATION PROCEDEURE"
$WRITE SYS$OUTPUT " INPT - GENERATE INPUT FILE TO PANEL GENERATION PROCE
$WRITE SYS$OUTPUT " MODD - MODIFY GEOMETRY FILES"
$WRITE SYS$OUTPUT " PLOT - PLOT GEOMETRY FILES"
$WRITE SYS$OUTPUT " LOUT - LOGOUT"
$WRITE SYS$OUTPUT " HELP - LIST AVAILABLE COMMANDS"
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
$GOTO LAB50
!
$HCPY:
$PLOT
$GOTO LAB50
!
$CFIL:
!
$SET NOON
$ASSIGN/USER_MODE SYS$COMMAND: SYS$INPUT:
$CREFIL 'NTERM'
$SET ON
$GOTO LAB50
!
$INPT:
!
$SET NOON
$ASSIGN/USER_MODE SYS$COMMAND: SYS$INPUT:
$SMHESS 'NTERM'
$IFL=$STATUS-%X1000000
$IF IFL .EQS. 1 THEN GOTO CON3          !GET GEOMETRY FILES
$IF IFL .EQS. 2 THEN GOTO CON2          !CREATE PROBLEM FILE FOR S-H RUN
$IF IFL .EQS. 3 THEN GOTO CON3          !ABORT SMHESS PROCEDEURE
$GOTO CON3
$CON2:
!
!       CREATE PROBLEM FILE
!
$P1=3
$GOTO LAB50
$CON3:
$P1=IFL
$GOTO LAB50
!
$MODD:
!
$SET NOON
$ASSIGN/USER_MODE SYS$COMMAND: SYS$INPUT:
$MODFY 'NTERM'
$P1=$STATUS-%X10000000
$SET ON
$IF P1 .EQS. 0 THEN GOTO LOUT          !EXIT WISH
$GOTO LAB50
!
$PLOT:
!$WRITE SYS!$OUTPUT " PLOT PACKAGE NOT CURRENTLY AVAILABLE ON NASA MACHI
!$GOTO LAB50
!
$ASSIGN SYS$INPUT FOR005
$ASSIGN SYS$OUTPUT FOR006
$IF NTERM .EQS. 1 THEN GOTO LAB80
$SET NOON
$ASSIGN/USER_MODE SYS$COMMAND: SYS$INPUT:
$PLTT 'NTERM' 1                        ! PLOT GEOMETRY FILES
$P1=$STATUS-%X10000000
$SET ON
$IF P1 .EQS. 0 THEN GOTO LOUT          !EXIT WISH
```

```
$GOTO LAB50
$LAB80:
$WRITE SYS$OUTPUT 'NO PLOTTING ON A VT100'
$GOTO LAB50
!
$EXEC:
$WRITE SYS$OUTPUT ' IF INPT COMMAND HAS BEEN EXECUTED JUST PRIOR'
$WRITE SYS$OUTPUT ' TO USING THIS COMMAND , THEN THE PROPER FILE '
$WRITE SYS$OUTPUT ' ASSIGNMENTS HAVE BEEN MADE. '
$WRITE SYS$OUTPUT '          '
$WRITE SYS$OUTPUT ' IF NOT , THEN THE ASSIGNMENTS HAVE TO BE MADE '
$WRITE SYS$OUTPUT ' NOW BY ANSWERING NO ( 1 = NO ) TO THE FOLLOWING '
$WRITE SYS$OUTPUT ' QUESTION . '
$WRITE SYS$OUTPUT '     '
$WRITE SYS$OUTPUT ' HAS INPT COMMAND BEEN EXECUTED JUST PRIOR TO '
$WRITE SYS$OUTPUT ' THIS COMMAND ? '
$WRITE SYS$OUTPUT '     '
$INQUIRE IASSGN ' ENTER 0 FOR YES OR 1 FOR NO'
$IF IASSGN .EQS. 0 THEN GOTO LAB65
$INQUIRE OTFILE1 ' NAME OF OUTPUT FILE FOR PANEL GEOMETRY TO BE USED IN
$ASSIGN 'OTFILE1' FOR004
$INQUIRE OTFILE2 ' OUTFILE FILE FOR BATCH PRINTING OF GEOMETRY ?'
$ASSIGN 'OTFILE2' FOR006
$INQUIRE INFILE ' FILE NAME FOR INPUT TO GEOMETRY PACKAGE ?'
$ASSIGN 'INFILE' FOR005
$ASSIGN SYS$OUTPUT FOR008
$LAB65:
$RUN GEOM
$GOTO LAB50
!
!
$PGEN:
$ASSIGN/USER_MODE SYS$COMMAND: SYS$INPUT:
$ASSIGN SYS$INPUT FOR005
$ASSIGN SYS$OUTPUT FOR006
$RUN PAFILE
$GOTO LAB50
!
$LOUT:
!
!       LOGOUT PROCEDURE
!
$IF NTERM .EQS. 2 THEN TKCLR
!
!       DELETE FILES TO CLEAN UP
!
!$LOGOUT
$EXIT
```

A.2.2 Code For Program SMHESS

The following pages contain the code developed for the SMHESS program.

```
C
C           PROGRAM SMHESS
C
C          PROGRAM SMHESS GENERATES INPUT FOR THE HALSEY/HESS
C          GEOMETRY PANELING PROCEDEURE
C
C
C          PROGRAMMMED ON 24 MARCH 1982              M.C.WITTE
C          ADDITIONS MADE ON MAY 7 1982              H. W. MANN
C
C
C
C
      DIMENSION INET(3,60),COORD(400,3,60)
      DIMENSION NB(60),NALGB(60),NW(60),NALGW(60),NS(60),NALGS(60),
     1          NALG2(60),NALG3(60),IXFLAG(60),IMODE(60),ITR(60),
     2          INTS(60)
      DIMENSION SSB(606,60),SSW(306,60),SSS(306,60)
      DIMENSION COSX(60),COSY(60),COSZ(60)
      DIMENSION NSEG(5),IEND(5,60)
      DIMENSION ITR2(5,60),TR1(5,60),TR2(5,60),TR3(5,60),TR4(5,60)
C
      CHARACTER*20 CMD,INPUT*4,TITLE*80,GNAME*13,INFILE*13,OUTPUT*49,
     1OTFILE1*13,OTFILE2*13,NTIT(60)*30,LINE*80
C
      LOGICAL*1 EX
      INTEGER STAT(400,2,60)
C
C            GET TERMINAL TYPE
C
      STATUS=LIB$GET_FOREIGN(INPUT,,N)
      READ(INPUT,800) NTERM
800   FORMAT(I1)
C
C             READ PRELIMINARY GEOMETRY FILE NAME
C
    5 WRITE(6,*)' ENTER PRELIMINARY GEOMETRY FILE NAME (13 CHARS)'
      IF(NTERM.EQ.3) WRITE(6,*)'  '
      IF(NTERM.EQ.4) WRITE(6,950)
      READ(5,810) GNAME
810   FORMAT(A)
C
C              TEST FOR VALID FILE NAME
C
      IF(GNAME(1:1).EQ.'Q') GO TO 400
      INQUIRE(FILE=GNAME,EXIST=EX)
      IF(.NOT.EX) THEN
        WRITE(6,*)' FILE DOES NOT EXIST'
        GO TO 5
      END IF
C
C              READ INPUT FROM PRELIMINARY GEOMETRY FILE
C
      OPEN(UNIT=1,FILE=GNAME,STATUS='OLD')
      READ(1,965) TITLE
965   FORMAT(A68)
      DO K=1,60
```

```
        READ(1,955,END=9)INET(3,K),INET(2,K),NTIT(K)
        INET(1,K)=K
        NNET=K
955     FORMAT(2I4,A30)
        WRITE(6,*)INET(3,K),INET(2,K)
        LEND=INET(2,K)*INET(3,K)
        DO J=1,LEND
        READ(1,960)COORD(J,1,K),COORD(J,2,K),COORD(J,3,K),STAT(J,1,K),
     1    STAT(J,2,K)
960       FORMAT(3F20.5,1X,I1,I1)
        END DO
      END DO
9     CLOSE(UNIT=1)
      NSEC=NNET
C
C             INITIAILIZE DEFAULT VALUES FOR OPTIONS AND VARIABLES
C
10    NOPT=1
      NPTS=1
      DO JJ=1,NSEC
      NB(JJ)=0
      NW(JJ)=0
      NS(JJ)=0
      NALGB(JJ)=0
      NALGW(JJ)=0
      NALGS(JJ)=0
      IMODE(JJ)=0
      NSEG(JJ)=1
      ITR(JJ)=0
      INTS(JJ)=0
      NALG2(JJ)=0
      NALG3(JJ)=0
      END DO
C
C
C       WRITE DEFAULT VALUES TO TERMINAL
C
C
      WRITE(6,*)'THE DEFAULT VALUES FOR INPUT TO THE PANEL GENERATION '
      WRITE(6,*)'PROCEDURE ARE :'
      WRITE(6,*)'   NOPT = 2 : PANEL ISOLATED COMPONENTS ONLY '
      WRITE(6,*)'   NB(1)= 0 : NUMBER OF ON-BODY POINTS PER N-LINE'
      WRITE(6,*)'              ON COMPONENT 1 '
      WRITE(6,*)'   NALGB(1)= 0 : SPACING ALGORITHM FOR ON-BODY POINTS'
      WRITE(6,*)'                 ON N-LINES FOR COMPONENT I - '
      WRITE(6,*)'                 INPUT DISTRIBUTION , UNALTERED '
      WRITE(6,*)'   NW(1)= 0 : NUMBER OF WAKE POINTS PER N-LINE ON '
      WRITE(6,*)'              COMPONENT 1 '
      WRITE(6,*)'   NALGW(1)= 0 : SPACING ALGORITHM FOR WAKE POINTS '
      WRITE(6,*)'                 PER N-LINE FOR COMPONENT I - '
      WRITE(6,*)'                 INPUT DISTRIBUTION , UNALTERED '
      WRITE(6,*)'   NS(1)= 0 : SPECIFIED NUMBER OF N- LINES ON COMP. 1 '
      WRITE(6,*)'   NALGS(1)= 0 : SPACING ALGORITHM FOR N-LINES ON '
      WRITE(6,*)'                 COMPONENT 1 - '
      WRITE(6,*)'                 INPUT DISTRIBUTION , UNALTERED '
      WRITE(6,*)'   INTS(1)= 0 : COMPONENT DOES NOT INTERSECT ANOTHER '
      WRITE(6,*)'                COMPONENT '
```

```
      WRITE(6,*)'     '
      WRITE(6,*)' PRESS "RETURN" TO CONTINUE'
      READ(5,*)
      WRITE(6,*)'  NALG2(I)= 0 : MINIMAL REPANELING '
      WRITE(6,*)'  NALG3(I)= 0 : NO REPANELING '
      WRITE(6,*)'  IMODE(I)= 0 : PLANAR- SECTION MODE USING DEFAULT '
      WRITE(6,*)'                  VALUES OF DIRECTION COSINES '
      WRITE(6,*)'  ITR(I)= 0 : NO COMPONENT TRANSFORMATIONS '
      WRITE(6,*)'  NSEG(I)= 1 : NUMBER OF SEGMENTS INTO WHICH M-LINES '
      WRITE(6,*)'                 ARE BROKEN '
      WRITE(6,*)'     '
C
C
C     USER HAS OPTION TO REVIEW ALL PROGRAM  OPTIONS AND VARIABLES AND/OR
C     ONLY THOSE SELECTED BY THE USER
C
C
      WRITE(6,*)'DO YOU WISH TO REVIEW INPUT OPTIONS AND VARIABLES FOR'
      WRITE(6,*)'ALL COMPONENTS ( YES OR NO ). '
      WRITE(6,*)'     '
      WRITE(6,*)'IF YOU SELECT ( NO ) , YOU WILL HAVE THE OPPORTUNITY '
      WRITE(6,*)'TO CHANGE ANY DEFAULT VALUES BY SPECIFING THE '
      WRITE(6,*)'PARAMETER AND COMPONENT TO BE CHANGED ; ONLY THOSE '
      WRITE(6,*)'SPECIFIED BY THE USER WILL BE ALTERED .     '
      WRITE(6,*)'     '
      WRITE(6,*)'IF YOU SELECT ( YES ) ,  YOU WILL HAVE THE OPPORTUNITY'
      WRITE(6,*)'TO REVIEW AND SELECT THE INPUT VARIABLES AND OPTIONS'
      WRITE(6,*)'FOR EACH COMPONENT  ; THEN THE USER WILL HAVE THE'
      WRITE(6,*)'OPPORTUNITY TO CHANGE ANY OPTION OR VARIABLE SELECTED'
      WRITE(6,*)'AS ABOVE'
      WRITE(6,*)'     '
      WRITE(6,*)'       ENTER Y FOR ( YES ) OR N FOR ( NO ) . '
      IF(NTERM.EQ.3) WRITE(6,*)'     '
      CALL YESNO(NY)
      IF(NY.EQ.0) GO TO 20
C
C
C        INPUT CONTROL AND REPANELING OPTIONS
C
C
C
      WRITE(6,*)'  NOPT < 2  , PANEL ISOLATED COMPONENTS ONLY (DEFAULT)'
      WRITE(6,*)'  NOPT = 2  , PANEL ISOLATED COMPONENTS AND CALCULATE'
      WRITE(6,*)'               INTERSECTION CURVES'
      WRITE(6,*)'  NOPT > 2  , PANEL ISOLATED COMPONENTS , CALCULATE '
      WRITE(6,*)'               INTERSECTION CURVES , AND REPANEL '
      WRITE(6,*)'               INTERSECTING AND INTERSECTED COMPONENTS'
      WRITE(6,*)'  ENTER AN INTERGER VALUE FOR NOPT , IE. 1, 2, OR 3'
      IF(NTERM.EQ.3) WRITE(6,*)'     '
      READ(5,*) NOPT
      DO II=1,NSEC
  705 FORMAT('  COMPONENT NO. ',I2,5X,A60)
      WRITE(6,*)'     '
      WRITE(6,*)'     '
      WRITE(6,*)'     '
      WRITE(6,*)'     '
      WRITE(6,705),II,NTIT(II)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
WRITE(6,*)'      '
WRITE(6,*)'ENTER INTERGER VALUE FOR NB(I) : SPECIFIED NUMBER OF'
WRITE(6,*)'ON-BODY POINTS PER N-LINE ON COMPONENT I '
IF(NTERM.EQ.3) WRITE(6,*)'      '
READ(5,*) NB(II)
WRITE(6,*)'ENTER INTERGER VALUE FOR NALGB(I) : SPACING ALGORITHM'
WRITE(6,*)'FOR ON-BODY POINTS PER N-LINE ON COMPONENT I '
WRITE(6,*)'      '
WRITE(6,*)'NALGB = 0 , INPUT DISTRIBUTION , UNALTERED (DEFAULT)'
WRITE(6,*)'NALGB = 1 , INPUT DISTRIBUTION , AUGMENTED '
WRITE(6,*)'NALGB = 2 , CONSTANT INCREMENTS IN ARC LENGTH '
WRITE(6,*)'NALGB = 3 , COSINE DISTRIBUTION '
WRITE(6,*)'NALGB = 4 , CURVATURE-DEPENDENT DISTRIBUTION '
WRITE(6,*)'NALGB = 5 , USER-SPECIFIED DISTRIBUTION '
WRITE(6,*)'     ENTER NALGB(I)   '
IF(NTERM.EQ.3) WRITE(6,*)'      '
READ(5,*) NALGB(II)
WRITE(6,*)'      '
WRITE(6,*)'ENTER A INTEGER VALUE FOR NW(I) : SPECIFIED NUMBER OF'
WRITE(6,*)'WAKE POINTS PER N-LINE ON COMPONENT I '
IF(NTERM.EQ.3) WRITE(6,*)'      '
READ(5,*) NW(II)
WRITE(6,*)'ENTER A INTEGER VALUE FOR NALGW(I) : SPACING ALGORITHM'
WRITE(6,*)'FOR WAKE POINTS ON N-LINE'
WRITE(6,*)'    '
WRITE(6,*)' NALGW = 0 , INPUT DISTRIBUTION , UNALTERED (DEFAULT)'
WRITE(6,*)' NALGW = 1 , INPUT DISTRIBUTION , AUGMENTED '
WRITE(6,*)' NALGW = 2 , CONSTANT INCREMENTS IN ARC LENGTH '
WRITE(6,*)'      '
WRITE(6,*)'              ENTER NALGW    '
IF(NTERM.EQ.3) WRITE(6,*)'      '
READ(5,*) NALGW(II)
WRITE(6,*)'      '
WRITE(6,*)'ENTER A INTEGER VALUE FOR NS(I) : SPECIFIED NUMBER OF'
WRITE(6,*)'N-LINES ON COMPONENT I '
IF(NTERM.EQ.3) WRITE(6,*)'      '
READ(5,*) NS(II)
WRITE(6,*)'      '
WRITE(6,*)'ENTER A INTEGER VALUE FOR NALGS(I) : SPACING ALGORITHM'
WRITE(6,*)'FOR N-LINES ON COMPONENT I '
WRITE(6,*)'      '
WRITE(6,*)' NALGS = 0 , INPUT DISTRIBUTION , UNALTERED (DEFAULT)'
WRITE(6,*)' NALGS = 1 , INPUT DISTRIBUTION , AUGMENTED '
WRITE(6,*)' NALGS = 2 , CONSTANT INCREMENTS IN ARC LENGTH '
WRITE(6,*)' NALGS = 3 , USER-SPECIFIED DISTRIBUTION '
WRITE(6,*)'      '
WRITE(6,*)'            ENTER NALGS    '
IF(NTERM.EQ.3) WRITE(6,*)'      '
READ(5,*) NALGS(II)
WRITE(6,*)'      '
WRITE(6,*)'ENTER A INTEGER VALUE FOR INTS(I) : INDEX OF THE'
WRITE(6,*)'COMPONENT (IF ANY) WHICH COMPONENT I INTERSECTS '
WRITE(6,*)'                 ENTER 0 IF NONE '
IF(NTERM.EQ.3) WRITE(6,*)'      '
READ(5,*) INTS(II)
WRITE(6,*)'      '
WRITE(6,*)'ENTER A INTEGER VALUE FOR NALG2(I) : REPANELING FLAG'
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      WRITE(6,*)'FOR INTERSECTING COMPONENTS ONLY '
      WRITE(6,*)'     NALG2 = 0 , MINIMAL REPANELING (DEFAULT) '
      WRITE(6,*)'     NALG2 = 1 , FULL REPANELING '
      WRITE(6,*)'     '
      WRITE(6,*)'             ENTER NALG2     '
      IF(NTERM.EQ.3) WRITE(6,*)'      '
      READ(5,*) NALG2(II)
      WRITE(6,*)'     '
      WRITE(6,*)'ENTER A INTEGER VALUE FOR NALG3(I) : REPANELING FLAG'
      WRITE(6,*)'FOR INTERSECTED COMPONENTS ONLY '
      WRITE(6,*)'    '
      WRITE(6,*)'NALG3 = 0 , NO REPANELING (DEFAULT) '
      WRITE(6,*)'NALG3 = 1 , (A) NONLIFTING COMPONENTS : '
      WRITE(6,*)'                FULL REPANELING WITH AN N-LINE PASSING'
      WRITE(6,*)'                THROUGH EVERY POINT ON INTERSECTION '
      WRITE(6,*)'                CURVE '
      WRITE(6,*)'            (B) LIFTING COMPONENTS : '
      WRITE(6,*)'                 FULL REPANELING BUT WITH GAPS IN THE'
      WRITE(6,*)'                 REGION OF THE INTERSECTION CURVE'
      WRITE(6,*)'NALG3 = 2 , (A) NONLIFTING COMPONENTS : '
      WRITE(6,*)'                FULL REPANELING WITH AN N-LINE PASSING'
      WRITE(6,*)'                THROUGH EVERY SECOND POINT ON THE'
      WRITE(6,*)'                INTERSECTION CURVE '
      WRITE(6,*)'            (B) LIFTING COMPONENTS : '
      WRITE(6,*)'                 FULL REPANELING WITH ADDITIONAL NON-'
      WRITE(6,*)'                 LIFTING ELEMENTS TO FILL THE GAPS'
      WRITE(6,*)'                 AROUND THE INTERSECTION CURVE '
      WRITE(6,*)'     '
      WRITE(6,*)'            ENTER NALG3     '
      IF(NTERM.EQ.3) WRITE(6,*)'      '
      READ(5,*) NALG3(II)
      WRITE(6,*)'     '
      WRITE(6,*)'ENTER A INTEGER VALUE FOR IXFLAG(I) : EXTRA STRIP FLAG'
      WRITE(6,*)'FOR COMPONENT I '
      WRITE(6,*)'     '
      WRITE(6,*)'IXFLAG = 0 , NO EXTRA STRIP (DEFAULT)'
      WRITE(6,*)'IXFLAG = 1 , THE FIRST STRIP OF THE FINAL COORDINATES'
      WRITE(6,*)'             IS EXTRA '
      WRITE(6,*)'IXFLAG = 3 , THE LAST STRIP OF THE FINAL COORDINATES'
      WRITE(6,*)'             IS EXTRA '
      WRITE(6,*)'     '
      WRITE(6,*)'                ENTER IXFLAG     '
      IF(NTERM.EQ.3) WRITE(6,*)'      '
      READ(5,*) IXFLAG(II)
      WRITE(6,*)'     '
      WRITE(6,*)'ENTER A INTEGER VALUE FOR IMODE(I) : PLANAR SECTION /'
      WRITE(6,*)'                              ARC-LENGTH INDICATOR '
      WRITE(6,*)'     '
      WRITE(6,*)'   IMODE = 0 , PLANAR SECTION MODE USING DEFAULT '
      WRITE(6,*)'               DIRECTION COSINES OF AXIS (1,0,0) FOR '
      WRITE(6,*)'               NONLIFTING COMPONENTS & (0,-1,0) FOR '
      WRITE(6,*)'               LIFTING COMPONENTS (DEFAULT) '
      WRITE(6,*)'   IMODE = 1 , PLANAR SECTION MODE USING INPUT VALUES'
      WRITE(6,*)'               OF DIRECTION COSINES OF AXIS '
      WRITE(6,*)'   IMODE = 2 , ARC LENGTH MODE '
      WRITE(6,*)'     '
      WRITE(6,*)'              ENTER IMODE    '
```

```
      IF(NTERM.EQ.3) WRITE(6,*)'      '
      READ(5,*) IMODE(II)
      WRITE(6,*)'   '
      WRITE(6,*)' ENTER AN INTEGER VALUE FOR ITR(I) : COMPONENT '
      WRITE(6,*)'                              TRANSFORMATION FLAG.'
      WRITE(6,*)'    '
      WRITE(6,*)' ( THE VALUE INDICATES THE NUMBER OF TRANSFORMATIONS'
      WRITE(6,*)'   TO BE PERFORMED - DEFAULT VALUE = 0 '
      WRITE(6,*)'          '
      WRITE(6,*)'          ENTER ITR    '
      IF(NTERM.EQ.3) WRITE(6,*)     '
      READ(5,*) ITR(II)
      WRITE(6,*)'    '
      WRITE(6,*)' ENTER AN INTEGER VALUE FOR NSEG(I) : NUMBER OF '
      WRITE(6,*)' SEGMENTS INTO WHICH M-LINES ARE BROKEN '
      WRITE(6,*)' (DEFAULT=1)'
      WRITE(6,*)'     '
      WRITE(6,*)'         ENTER NSEG '
      IF(NTERM.EQ.3) WRITE(6,*)'    '
      READ(5,*) NSEG(II)
      WRITE(6,*)'     '
      WRITE(6,*)'    '
      WRITE(6,*)'    '
      WRITE(6,*)'    '
      WRITE(6,707) II,NTIT(II)
  707 FORMAT(' OPTIONS FOR COMPONENT NO. ',I2,5X,A60)
  708 FORMAT(' NB=',I3,2X,'NALGB=',I2,2X,'NW=',I2,2X,'NALGW=',I2)
      WRITE(6,*)'    '
      WRITE(6,708) NB(II),NALGB(II),NW(II),NALGW(II)
  709 FORMAT(' NS=',I3,2X,'NALGS=',I2,2X,'INTS=',I2,2X,'NALG2=',I2)
      WRITE(6,*)'    '
      WRITE(6,709) NS(II),NALGS(II),INTS(II),NALG2(II)
  710 FORMAT(' NALG3=',I2,2X,'IXFLAG=',I2,2X,'IMODE=',I2)
      WRITE(6,*)'    '
      WRITE(6,710) NALG3(II),IXFLAG(II),IMODE(II)
  719 FORMAT(' ITR=',I2,2X,'NSEG=',I2)
      WRITE(6,*)'    '
      WRITE(6,719) ITR(II),NSEG(II)
      WRITE(6,*)'    '
      END DO
      DO II=1,NSEC
      IF(NALGB(II).EQ.5) THEN
      WRITE(6,*)'    '
      WRITE(6,*)'    '
      WRITE(6,*)'ADDITIONAL INPUT REQUIRED FOR :'
  933 FORMAT('    COMPONENT NO. ',I2,5X,A)
      WRITE(6,933) II,NTIT(II)
      WRITE(6,*)'    '
      WRITE(6,*)' ENTER NB(I) VALUES OF SSB(J,I) ,  J=1,NB(I) '
      WRITE(6,*)'   SSB : SPECIFIED ARC LENGTHS OF ON BODY POINTS ON'
      WRITE(6,*)'         N-LINES ON COMPONENT I ( 0.0 - SSB - 1.0 )'
      WRITE(6,*)'   '
      DO JJ=1,NB(II)
      WRITE(6,*)'    '
      WRITE(6,711) II,JJ
  711 FORMAT(' ENTER VALUE OF SSB FOR COMPONENT NO. ',I2,2X,'POINT NO.',
     1       I2)
```

```
      IF(NTERM.EQ.3) WRITE(6,*)'     '
      READ(5,*) SSB(JJ,II)
      END DO
      END IF
      IF(NALGW(II).EQ.5) THEN
      WRITE(6,*)'     '
      WRITE(6,*)'ADDITIONAL INPUT REQUIRED FOR :'
      WRITE(6,933) II,NTIT(II)
      WRITE(6,*)'     '
      WRITE(6,*)' ENTER NW(I) VALUES OF SSW(J,I) ,  J=1,NW(I) '
      WRITE(6,*)'   SSW : SPECIFIED ARC LENGTHS OF WAKE POINTS ON '
      WRITE(6,*)'         N-LINES ON COMPONENT I ( 0.0 - SSW - 1.0 )'
      DO JJ=1,NW(II)
      WRITE(6,*)'     '
      WRITE(6,717) II,JJ
  717 FORMAT(' ENTER VALUE OF SSW FOR COMPONENT NO. ',I2,2X,'POINT NO.',
     1       I2)
      IF(NTERM.EQ.3) WRITE(6,*)'     '
      READ(5,*) SSW(JJ,II)
      END DO
      END IF
      IF(NALGS(II).EQ.3) THEN
      WRITE(6,*)'     '
      WRITE(6,*)' ADDITIONAL INPUT REQUIRED FOR : '
332   FORMAT('   COMPONENT NO.  ',I2,5X,A)
      WRITE(6,332) II,NTIT(II)
      WRITE(6,*)'     '
      WRITE(6,*)'  ENTER NS(I) VALUES OF SSS(J,I) '
      WRITE(6,*)'     ( J = 1 , NS(I) ) '
      WRITE(6,*) '     '
      WRITE(6,*)'   SSS: SPECIFIED DISTRIBUTION OF N-LINES'
      WRITE(6,*)'        ON COMPONENT I ( 0.0 - SSS - 1.0 ).'
      WRITE(6,*)'     '
      DO JJ=1,NS(II)
      WRITE(6,*)'     '
      WRITE(6,333) II,JJ
333   FORMAT(' ENTER VALUE FOR SSS FOR COMPONENT NO. ',I2,
     2 '   POINT NO. ',I2)
      IF(NTERM.EQ.3) WRITE(6,*)'     '
      READ(5,*) SSS(JJ,II)
      END DO
      END IF
      IF(IMODE(II).EQ.1) THEN
      WRITE(6,*)'     '
      WRITE(6,*)'ADDITIONAL INPUT REQUIRED FOR :'
      WRITE(6,933) II,NTIT(II)
      WRITE(6,*)'     '
      WRITE(6,*)' ENTER DIRECTION COSINES : '
      WRITE(6,*)'   COSX(I) - X-DIRECTION COSINE OF THE AXIS OF COMP. I'
      WRITE(6,*)'   COSY(I) - Y-DIRECTION COSINE OF THE AXIS OF COMP. I'
      WRITE(6,*)'   COSZ(I) - Z-DIRECTION COSINE OF THE AXIS OF COMP. I'
      WRITE(6,*)'     '
  712 FORMAT(' ENTER VALUE OF COSX(I) FOR COMPONENT NO. ',I2)
      WRITE(6,712) II
      IF(NTERM.EQ.3) WRITE(6,*)'     '
      READ(5,*) COSX(II)
  713 FORMAT(' ENTER VALUE OF COSY(I) FOR COMPONENT NO. ',I2)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      WRITE(6,*)'   '
      WRITE(6,712) II
      IF(NTERM.EQ.3) WRITE(6,*)'    '
      READ(5,*) COSY(II)
  714 FORMAT(' ENTER VALUE OF COSZ(I) FOR COMPONENT NO. ',I2)
      WRITE(6,*)'   '
      WRITE(6,712) II
      IF(NTERM.EQ.3) WRITE(6,*)'    '
      READ(5,*) COSZ(II)
      END IF
      IF(NSEG(II).GT.1) THEN
      WRITE(6,*)'   '
      WRITE(6,*)'ADDITIONAL INPUT REQUIRED FOR :'
      WRITE(6,933) II,NTIT(II)
      WRITE(6,*)'   '
      WRITE(6,*)'ENTER NSEG(I) VALUES OF IEND(J,I) ,  J=1,NSEG(I)'
      WRITE(6,*)' IEND : POINT NUMBERS OF THE ENDS OF THE SEGMENTS'
      WRITE(6,*)'        ON M-LINES ON COMPONENT I .'
      WRITE(6,*)'   '
      WRITE(6,*)' NOTE : NSEG(I) TOTAL (MAX.=5) PER COMPONENT ONLY'
      WRITE(6,*)'        REQUIRED IF NSEG(I) > 1 . '
      DO JJ=1,NSEG(II)
      WRITE(6,*)'   '
      WRITE(6,715) JJ
  715 FORMAT(' ENTER VALUE OF IEND FOR SEGMENT NO. ',I2)
      IF(NTERM.EQ.3) WRITE(6,*)'    '
      READ(5,*) IEND(JJ,II)
      END DO
      END IF
      IF(ITR(II).GT.0) THEN
      WRITE(6,*)'   '
      WRITE(6,*)'ADDITIONAL INPUT REQUIRED FOR :'
      WRITE(6,933) II,NTIT(II)
      WRITE(6,*)'   '
      WRITE(6,*)'ENTER TRANSFORMATION PARAMETERS - A TOTAL OF ITR(I)'
      WRITE(6,*)'TIMES FOR COMPONENT I   '
      WRITE(6,*)'EACH TRANSFORMATION REQUIRES THE INPUT OF ITR2 , TR(1)'
      WRITE(6,*)', TR(2) , TR(3) , AND TR(4)    '
      WRITE(6,*)'   '
      WRITE(6,*)' ITR2 : INDICATOR FOR TYPE OF TRANSFORMATION'
      WRITE(6,*)'             = 1 , SCALING        '
      WRITE(6,*)'             = 2 , TRANSLATION    '
      WRITE(6,*)'             = 3 , ROTATION       '
      WRITE(6,*)'          '
      WRITE(6,*)' TR(1) : FIRST TRANSFORMATION PARAMETER - '
      WRITE(6,*)'       MULTIPLICATION FACTOR ON X-SCALE IF ITR2 = 1'
      WRITE(6,*)'       TRANSLATION OF X-SCALE            IF ITR2 = 2'
      WRITE(6,*)'       ANGLE OF ROTATION                 IF ITR2 = 3'
      WRITE(6,*)'           (POSITIVE CLOCKWISE , LOOKING DOWN AXIS'
      WRITE(6,*)'                  AWAY FROM ORIGIN)'
      WRITE(6,*)' TR(2) : SECOND TRANSFORMATION PARAMETER -  '
      WRITE(6,*)'       MULTIPLICATION FACTOR ON Y-SCALE IF ITR2 = 1'
      WRITE(6,*)'       TRANSLATION OF Y-SCALE            IF ITR2 = 2'
      WRITE(6,*)'       X DIRECTION COSINE OF AXIS OF ROTATION '
      WRITE(6,*)'                                         IF ITR2 = 3'
      WRITE(6,*)' TR(3) : THIRD TRANSFORMATION PARAMETER -
      WRITE(6,*)'       MULTIPLICATION FACTOR ON Z-SCALE IF ITR2 = 1'
```

```
      WRITE(6,*)'         TRANSLATION OF Z-SCALE              IF ITR2 = 2'
      WRITE(6,*)'         Y DIRECTION COSINE OF AXIS OF ROTATION '
      WRITE(6,*)'                                             IF ITR2 = 3'
      WRITE(6,*)' TR(4) : FOURTH TRANSFORMATION PARAMETER -        '
      WRITE(6,*)'          DUMMY IF ITR2 = 1 OR ITR2 = 2  '
      WRITE(6,*)'          Z DIRECTION COSINE OF AXIS OF ROTATION '
      WRITE(6,*)'                IF ITR2 = 3'
      WRITE(6,*)'        '
      WRITE(6,*)'NOTE:'
      WRITE(6,*)' THE AXIS OF ROTATION IS ASSUMED TO PASS THROUGH THE'
      WRITE(6,*)' ORIGIN . THIS INPUT IS REQUIRED IF ITR(I) IS NOT'
      WRITE(6,*)' EQUAL TO 0 FOR ANY COMPONENT I . A TOTAL OF ITR(I)'
      WRITE(6,*)' TRANSFORMATIONS ARE PERFORMED FOR EACH COMPONENT I .'
      DO JJ=1,ITR(II)
      WRITE(6,*)'   '
      WRITE(6,*)'   '
      WRITE(6,716) II,JJ
  716 FORMAT('   COMPONENT NO. ',I2,2X,'  TRANSFORMATION NO. ',I2)
      WRITE(6,*)'    '
      WRITE(6,*)'  ENTER VALUE FOR ITR2   '
      IF(NTERM.EQ.3) WRITE(6,*)'    '
      READ(5,*) ITR2(JJ,II)
      WRITE(6,*)'    '
      WRITE(6,*)'  ENTER VALUE FOR TR(1)   '
      IF(NTERM.EQ.3) WRITE(6,*)'    '
      READ(5,*) TR1(JJ,II)
      WRITE(6,*)'    '
      WRITE(6,*)'  ENTER VALUE FOR TR(2)   '
      IF(NTERM.EQ.3) WRITE(6,*)'    '
      READ(5,*) TR2(JJ,II)
      WRITE(6,*)'    '
      WRITE(6,*)'  ENTER VALUE FOR TR(3)   '
      IF(NTERM.EQ.3) WRITE(6,*)'    '
      READ(5,*) TR3(JJ,II)
      WRITE(6,*)'    '
      WRITE(6,*)'  ENTER VALUE FOR TR(4)   '
      IF(NTERM.EQ.3) WRITE(6,*)'    '
      READ(5,*) TR4(JJ,II)
      WRITE(6,*)'    '
      END DO
      END IF
      END DO
C
C           WRITE PROMPT FOR VARIABLE SPECIFICATION
C
      WRITE(6,*)'   '
      WRITE(6,*)' YOU WILL NOW HAVE THE OPPORTUNITY TO CHANGE'
      WRITE(6,*)' ANY COMPONENT VARIABLE OR OPTION BY SPECIFING'
      WRITE(6,*)' THE VARIABLE AND THE COMPONENT NO.'
      WRITE(6,*)'    '
      WRITE(6,*)'   '
      WRITE(6,*)'   '
20    WRITE(6,*)' ENTER VARIABLE NAME AND VALUE'
      WRITE(6,*)'        EG.      VAR(1)=3.5 '
      WRITE(6,*)' ENTER "RESET" TO RESET THE DEFAULT VALUES'
      WRITE(6,*)' ENTER "Q" WHEN FINISHED'
      WRITE(6,*)' ENTER "ABORT" TO TERMINATE'
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      IF(NTERM.EQ.3) WRITE(6,*)'   '
      IF(NTERM.EQ.4) WRITE(6,950)
950   FORMAT(/)
30    WRITE(6,*)' ENTER COMMAND'
      IF(NTERM.EQ.3) WRITE(6,*)'   '
      IF(NTERM.EQ.4) WRITE(6,950)
      READ(5,810,ERR=20)CMD
C
C                TEST FOR ABORT
C
      IF(CMD(1:1).EQ.'A') GO TO 400
C
C                TEST FOR END OF INPUT
C
      IF(CMD(1:1).EQ.'Q') GO TO 300
C
C                 TEST FOR RESET DEFAULT
C
      IF(CMD(1:1).EQ.'R') GO TO 10
C
C                 TEST FOR HELP REQUEST
C
      IF(CMD(1:1).EQ.'H') THEN
C
C            LIST COMMANDS
C
      WRITE(6,*)' DO YOU WISH A PRINTED LISTING OF THE VARIABLES?'
      IF(NTERM.EQ.3) WRITE(6,*)'   '
      IF(NTERM.EQ.4) WRITE(6,950)
      CALL YESNO(NY)
      IF(NY.EQ.1) THEN
        OPEN(UNIT=1,FILE='SMHSINFO.SAV',STATUS='OLD',DISP='PRINT')
        CLOSE(UNIT=1)
        WRITE(6,*)' OBTAIN LISTING AT VAX VERSATEC'
      ELSE
C
C           MAKE SMALL LETTERS ON 4014
C
        IF(NTERM.EQ.3) THEN
          CALL H4014(0)
        END IF
        OPEN(UNIT=1,FILE='SMHSINFO.SAV',STATUS='OLD')
        IOS=0
        I=0
        DOWHILE (IOS.EQ.0)
          READ(1,810,IOSTAT=IOS) LINE
          IF(IOS.EQ.0) THEN
            I=I+1
            IF(I.EQ.22) THEN
              WRITE(6,*)' ENTER <CR> TO CONTINUE'
              READ(5,*)
              I=0
            END IF
            WRITE(6,820) LINE
          END IF
820       FORMAT(1X,A)
        END DO
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
        CLOSE(UNIT=1)
C
C            MAKE BIG LETTERS ON 4014
C
        IF(NTERM.EQ.3) THEN
          CALL H4014(2)
        END IF
      END IF
      GO TO 30
      END IF
C
C                   DECODE COMMAND INPUT
C
      CALL DECODE(CMD,NSEC,ICMD,ICOMP,VAL,IVAL,IERR,ICOMP2)
C
C             CHECK FOR COMMAND INPUT ERROR
C
      IF(IERR.NE.0) THEN
        IF(IERR.EQ.1) WRITE(6,*)' INVALID VARIABLE NAME ',CMD
        IF(IERR.EQ.2) WRITE(6,*)' INVALID ARGUMENT SPEC. ',CMD
        IF(IERR.EQ.3) WRITE(6,*)' INVALID COMPONENT SPEC ',CMD
        IF(IERR.EQ.4) WRITE(6,*)' INVALID VALUE SPEC. ',CMD
        GO TO 30
      END IF
C
C                BRANCH TO ASSIGN APPROPRIATE VARIABLE
C
      GO TO (40,50,60,70,80,90,100,110,120,130,140,150,
     1160,170,180,190,200,210,220,230,240,250,260,270
     2)ICMD
      WRITE(6,*)' INVALID COMMAND',CMD
C
C                VARIBLE NSEC
C
40    NSEC=IVAL
      GO TO 30
C
C                VARIABLE NOPT
C
50    NOPT=IVAL
      GO TO 30
C
C                VARIABLE NB
C
60    NB(ICOMP)=IVAL
      GO TO 30
C
C                VARIABLE NALGB
C
70    NALGB(ICOMP)=IVAL
      GO TO 30
C
C                VARIBLE NW
C
80    NW(ICOMP)=IVAL
      GO TO 30
C
```

```
C                VARIABLE NALGW
C
90    NALGW(ICOMP)=IVAL
      GO TO 30
C
C                VARIABLE NS
C
100   NS(ICOMP)=IVAL
      GO TO 30
C
C                VARIABLE NALGS
C
110   NALGS(ICOMP)=IVAL
      GO TO 30
C
C                VARIABLE INTS
C
120   INTS(ICOMP)=IVAL
      GO TO 30
C
C                VARIABLE NALG2
C
130   NALG2(ICOMP)=IVAL
      GO TO 30
C
C                VARIABLE NALG3
C
140   NALG3(ICOMP)=IVAL
      GO TO 30
C
C                VARIABLE IXFLAG
C
150   IXFLAG(ICOMP)=IVAL
      GO TO 30
C
C                VARIABLE IMODE
C
160   IMODE(ICOMP)=IVAL
      GO TO 30
C
C                VARIABLE ITR
C
170   ITR(ICOMP)=IVAL
      GO TO 30
C
C                VARIABLE NSEG
C
180   NSEG(ICOMP)=IVAL
      GO TO 30
C
C                VARIABLE COSX
C
190   COSX(ICOMP)=VAL
      GO TO 30
C
C                VARIABLE COSY
C
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
200   COSY(ICOMP)=VAL
      GO TO 30
C
C             VARIABLE COSZ
C
210   COSZ(ICOMP)=VAL
      GO TO 30
C
C             VARIABLE SSB
C
220   SSB(ICOMP2,ICOMP)=VAL
      GO TO 30
C
C             VARIABLE SSW
C
230   SSW(ICOMP2,ICOMP)=VAL
      GO TO 30
C
C             VARIABLE SSS
C
240   SSS(ICOMP2,ICOMP)=VAL
      GO TO 30
C
C             VARIABLE IEND
C
250   IEND(ICOMP2,ICOMP)=VAL
      GO TO 30
C
C             VARIABLE ITR2
C
260   ITRANS=1
      WRITE(6,*)'    '
      WRITE(6,*)'IS THERE MORE THAN ONE TRANSFORMATION TO BE DEFINED'
      WRITE(6,*)'FOR THIS COMPONENT ? '
      WRITE(6,*)'        ENTER Y FOR YES OR N FOR NO   '
      IF(NTERM.EQ.3) WRITE(6,*)'     '
      CALL YESNO(NY)
      IF(NY.EQ.1) THEN
      WRITE(6,*)'ENTER THE TRANSFORMATION NUMBER FOR WHICH THIS'
      WRITE(6,*)'VARIABLE APPLIES , IE. 1,2,3,4 OR 5  (MAX. = 5)'
      IF(NTERM.EQ.3) WRITE(6,*)'    '
      READ(5,*) ITRANS
      END IF
      WRITE(6,*)'ENTER THE COMPONENT NUMPER TO  WHICH THIS'
      WRITE(6,*)'TRANSFORMATION WILL BE APPLIED'
      WRITE(6,*)'    ( ENTER A INTERGER VALUE ) '
      IF(NTERM.EQ.3) WRITE(6,*)'    '
      READ(5,*) JCOMP
      ITR2(ITRANS,JCOMP)=VAL
      GO TO 30
C
C             VARIABLE TR
C
270   ITRANS=1
      WRITE(6,*)'    '
      WRITE(6,*)'IS THERE MORE THAN ONE TRANSFORMATION TO BE DEFINED'
      WRITE(6,*)'FOR THIS COMPONENT ? '
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      WRITE(6,*)'        ENTER Y FOR YES OR N FOR NO   '
      IF(NTERM.EQ.3) WRITE(6,*)'     '
      CALL YESNO(NY)
      IF(NY.EQ.1) THEN
      WRITE(6,*)'ENTER THE TRANSFORMATION NUMBER FOR WHICH THIS'
      WRITE(6,*)'VARIABLE APPLIES , IE. 1,2,3,4 OR 5  (MAX. = 5)'
      IF(NTERM.EQ.3) WRITE(6,*)'    '
      READ(5,*) ITRANS
      END IF
      WRITE(6,*)'ENTER THE COMPONENT NUMBER TO  WHICH THIS'
      WRITE(6,*)'TRANSFORMATION WILL BE APPLIED'
      WRITE(6,*)'  ENTER AN INTERGER VALUE'
      IF(NTERM.EQ.3) WRITE(6,*)'    '
      READ(5,*) JCOMP
      IF(ICOMP.EQ.1) TR1(ITRANS,JCOMP)=VAL
      IF(ICOMP.EQ.2) TR2(ITRANS,JCOMP)=VAL
      IF(ICOMP.EQ.3) TR3(ITRANS,JCOMP)=VAL
      IF(ICOMP.EQ.4) TR4(ITRANS,JCOMP)=VAL
      GO TO 30
C
C             ENTER INPUT FILE NAME
C
300   WRITE(6,*)' ENTER INPUT FILE NAME (13 CHARS)'
      WRITE(6,*)'       THIS WILL BE  THE INPUT FILE TO THE '
      WRITE(6,*)'       PANEL GENERATION PROGRAM'
      WRITE(6,*)'INPUT FILE NAME ? '
      IF(NTERM.EQ.3) WRITE(6,*)'   '
      IF(NTERM.EQ.4) WRITE(6,950)
      READ(5,810) INFILE
      WRITE(6,*)'    '
C
C              WRITE THE INPUT FILE
C
      OPEN(UNIT=2,FILE=INFILE,STATUS='NEW')
C
C           WRITE FILE TITLE
C
      WRITE(2,810)TITLE
C
C            WRITE OPTIONS
C
      WRITE(2,811) NSEC,NPTS,NOPT
  811 FORMAT(1X,I2,1X,I1,1X,I1)
      DO II=1,NSEC
      WRITE(2,812) NB(II),NALGB(II),NW(II),NALGW(II),NS(II),
     1      NALGS(II),INTS(II),NALG2(II),NALG3(II),
     2      IXFLAG(II),IMODE(II),ITR(II),NSEG(II),NTIT(II)
  812 FORMAT(1X,I2,1X,I1,1X,I2,1X,I1,1X,I3,1X,I1,1X,I2,1X,
     1       I1,1X,I1,1X,I1,1X,I1,1X,I1,1X,I1,1X,A30)
      END DO
      DO II=1,NSEC
      IF(IMODE(II).EQ.1) THEN
      WRITE(2,813) COSX(II),COSY(II),COSZ(II)
  813 FORMAT(3F10.5)
      END IF
      IF(NALGB(II).EQ.5) THEN
      DO JJ=1,NB(II)
```

```
      WRITE(2,814) SSB(JJ,II)          ORIGINAL PAGE IS
  814 FORMAT(F10.5)                     OF POOR QUALITY
      END DO
      END IF
      IF(NALGW(II).EQ.5) THEN
      DO JJ=1,NW(II)
      WRITE(2,815) SSW(JJ,II)
  815 FORMAT(F10.5)
      END DO
      END IF
      IF(NALGS(II).EQ.3) THEN
      DO JJ=1,NS(II)
      WRITE(2,816) SSS(JJ,II)
  816 FORMAT(F10.5)
      END DO
      END IF
      IF(NSEG(II).GT.1) THEN
      DO JJ=1,NSEG(II)
      WRITE(2,818) IEND(JJ,II)
  818 FORMAT(1I4)
      END DO
      END IF
      IF(ITR(II).GT.0) THEN
      DO JJ=1,ITR(II)
      WRITE(2,817) ITR2(JJ,II),TR1(JJ,II),TR2(JJ,II),
     1             TR3(JJ,II),TR4(JJ,II)
  817 FORMAT(I1,4F10.5)
      END DO
      END IF
      END DO
C
C            WRITE NETWORK DATA
C
      DO K=1,NNET
        LEND=INET(3,K)*INET(2,K)
        DO J=1,LEND
          WRITE(2,960)COORD(J,1,K),COORD(J,2,K),COORD(J,3,K),
     1           STAT(J,1,K),STAT(J,2,K)
        END DO
      END DO
      CLOSE(UNIT=2)
C
C             ENTER OUTPUT FILE NAMES
C
      WRITE(6,*)'     '
      WRITE(6,*)' ENTER OUTPUT FILE NAME (13 CHARS)'
      WRITE(6,*)'     THIS WILL BE THE OUTPUT FILE '
      WRITE(6,*)'     FROM THE PANEL GENERATION PROGRAM'
      WRITE(6,*)'     WHICH WILL BE USED FOR PLOTTING '
      WRITE(6,*)'     RESULTS AND GENERATING PANAIR'
      WRITE(6,*)'     INPUT'
      WRITE(6,*)'OUTPUT FILE NAME ? '
      IF(NTERM.EQ.3) WRITE(6,*)'   '
      IF(NTERM.EQ.4) WRITE(6,950)
      READ(5,810) OTFILE1
      WRITE(6,*)'     '
      WRITE(6,*)' ENTER BATCH OUTPUT FILE NAME (13 CHARS)'
```

```
      WRITE(6,*)'     THIS WILL BE THE OUTPUT FILE '
      WRITE(6,*)'     FROM THE PANEL GENERATION PROGRAM'
      WRITE(6,*)'     WHICH WILL BE PRINTED IF A COPY OF
      WRITE(6,*)'     THE INPUT AND RESULTS ARE DESIRED'
      WRITE(6,*)'OUTPUT FILE NAME ? '
      IF(NTERM.EQ.3) WRITE(6,*)'  '
      IF(NTERM.EQ.4) WRITE(6,950)
      READ(5,810) OTFILE2
C
C               RUN COMMAND FILE TO ASSIGN FILES TO LUN'S
C
      WRITE(OUTPUT,910)INFILE,OTFILE1,OTFILE2
910   FORMAT('$@ASSGN ',A,A,A)
      STATUS=LIB$DO_COMMAND(OUTPUT)
C
C               PROGRAM ABORT
C
400   WRITE(OUTPUT,920) 3
920   FORMAT('$@TRANSF ',I1)
      STATUS=LIB$DO_COMMAND(OUTPUT)
      END
C
C
C

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
C
      SUBROUTINE DECODE(CMD,NSEC,ICMD,ICOMP,VAL,IVAL,IERR,ICOMP2)
C
C            DECODE INTERPRETS AN INPUT STRING FOR
C          PORCEDEURE SMHESS
C
C
C        PROGRAMMED ON 24 MARCH 1982            M.C. WITTE
C
C
      CHARACTER*6 VARNAM(24),CMD*20
      DATA VARNAM/'NSEC  ','NOPT  ','NB    ','NALGB ','NW    '
     1           ,'NALGW ','NS    ','NALGS ','INIS  ','NALG2 '
     2           ,'NALG3 ','IXFLAG','IMODE ','ITR   ','NSEG  '
     3           ,'COSX  ','COSY  ','COSZ  ','SSB   ','SSW   '
     4           ,'SSS   ','IEND  ','ITR2  ','IR    '/
C
      IERR=0
C
C               FIND NUMBER OF CHARACTERS IN VARIABLE NAME
C
      NCHR3=0
      NCHR=1
      DOWHILE (CMD(NCHR:NCHR).NE.'('.AND.CMD(NCHR:NCHR).NE.'=')
        NCHR=NCHR+1
C
C               INVALID VARIABLE NAME
```

```
C
          IF(NCHR.GT.7) THEN
            ICMD=0
            ICOMP=0
            IERR=1
            RETURN
          END IF
        END DO
        NCHR=NCHR-1
C
C                    INVALID VARIBLE NAME
C
        IF(NCHR.LT.2) THEN
          ICMD=0
          ICOMP=0
          IERR=1
          RETURN
        END IF
C
C                    DETERMINE SELECTED VARIBLE NAME
C
        ICMD=1
        DOWHILE (CMD(1:NCHR).NE.VARNAM(ICMD)(1:NCHR))
          ICMD=ICMD+1
C
C                    VARIABLE NAME NOT FOUND
C
          IF(ICMD.GT.24) THEN
            ICMD=0
            ICOMP=0
            IERR=1
            RETURN
          END IF
        END DO
C
C
C     CHECK FOR COMMANDS WHICH ARE SHORTENED VERSIONS OF A
C     SIMILARLY LABELED COMMAND
C
C
        IF(CMD(1:NCHR).EQ.'NS') ICMD=7
        IF(CMD(1:NCHR).EQ.'ITR') ICMD=14
C
C
C                 CHECK FOR NSEC OR NOPT
C
C
        IF(ICMD.LT.3) THEN
        NCHR2=NCHR
        GO TO 351
        END IF
C
C                   DOES VARIBLE NAME HAVE AN ARGUMENT
C
        NCHR=NCHR+1
        IF(CMD(NCHR:NCHR).EQ.' ') THEN
          ICOMP=0
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
        IERR=2
        RETURN
      END IF
C
C                  VARIABLE NAME HAS AN ARGUMENT
C
      NCHR2=NCHR+1
      DOWHILE (CMD(NCHR2:NCHR2).NE.')'.AND.CMD(NCHR2:NCHR2).NE.',')
        NCHR2=NCHR2+1
C
C                  INVALID ARGUMENT SPECIFICATION
C
        IF(NCHR2.GT.10) THEN
          ICMD=0
          ICOMP=0
          IERR=2
          RETURN
        END IF
      END DO
C
C             CHECK FOR DOUBLE ARGUMENT
C
      IF(CMD(NCHR2:NCHR2).EQ.',') THEN
      NCHR3=NCHR2
      DOWHILE (CMD(NCHR3:NCHR3).NE.')')
      NCHR3=NCHR3+1
C
C               CHECK FOR INVALID ARGUMENT SPECIFICATION
C
      IF(NCHR3.GT.10) THEN
      ICMD=0
      ICOMP=0
      IERR=2
      RETURN
      END IF
      END DO
      END IF
C
C                 GET FIRST ARGUMENT (DOUBLE ARGUMENT SPECIFIED)
C
      IF(NCHR3.GT.0) THEN
      NCHR2=NCHR2+1
      NCHR3=NCHR3-1
      IF(NCHR3.EQ.NCHR2) THEN
      READ(CMD(NCHR2:NCHR2),900) ICOMP2
      ELSE
      READ(CMD(NCHR2:NCHR3),911) ICOMP2
      END IF
      NCHR2=NCHR2-1
      END IF
C
C                  INVALID ARGUMENT SPECIFICATION
C
      IF(NCHR2-NCHR.LE.1) THEN
        ICMD=0
        ICOMP=0
        IERR =2
```

```
        RETURN
      END IF
C
C                    GET COMPONENT NUMBER
C
      NCHR2=NCHR2-1
      NCHR=NCHR+1
      IF(NCHR.EQ.NCHR2) THEN
        READ(CMD(NCHR:NCHR),900)ICOMP
900     FORMAT(I1)
      ELSE
        READ(CMD(NCHR:NCHR2),911) ICOMP
911     FORMAT(I2)
      END IF
C
C                    TEST FOR PROPER COMPONENT RANGE
C
      IF(ICMD.NE.24) THEN
        IF(ICOMP.GT.NSEC.OR.ICOMP.LT.0) THEN
          ICMD=0
          ICOMP=0
          IERR=3
          RETURN
        END IF
      ELSE
        IF(ICOMP.GT.4.OR.ICOMP.LT.1) THEN
          ICMD=0
          ICOMP=0
          IERR=3
          RETURN
        END IF
      END IF
C
C                    DECODE VAL/IVAL
C
  351 NCHR2=NCHR2+1
      DOWHILE (CMD(NCHR2:NCHR2).NE.'=')
        NCHR2=NCHR2+1
        IF(NCHR2.GT.20) THEN
          ICMD=0
          ICOMP=0
          IERR=4
          RETURN
        END IF
      END DO
      NCHR2=NCHR2+1
C
C                GET REAL VALUE
C
      IF(ICMD.EQ.24) READ(CMD(NCHR2:20),921) VAL
921   FORMAT(F9.5)
      IF(ICMD.GE.16.AND.ICMD.LE.21) READ(CMD(NCHR2:20),921) VAL
C
C                GET INTEGER VALUE
C
      IF( ICMD.LT.16) READ(CMD(NCHR2:20),930) IVAL
930   FORMAT(BN,I3)
```

```
      IF(ICMD.GT.21.AND.ICMD.LT.24) READ(CMD(NCHR2:20),930) IVAL
C
      RETURN
      END
C ******************
C
C SUBROUTINE YESNO
C
C ******************
C
C PROGRAMMER: M.C. WITTE
C
C YESNO READS THE USERS ANSWER TO A QUESTION (EITHER Y (FOR YES)
C , N (FOR NO)), OR Q (FOR QUIT) AND SENDS A 1 , 0,OR -1
C (FOR Y , N,OR Q), BACK TO THE CALLING ROUTINE.
C
C NOMENCLATURE:
C
C N       - ALPHANUMERIC NO
C NY      - NUMERIC ANSWER(1 OR 0)
C Y       - ALPHANUMERIC YES
C YN      - USER ANSWER TO QUESTION
C Q       - ALPHANUMERIC QUIT
C
C
      SUBROUTINE YESNO(NY)
      CHARACTER*1 YN, YES*1/'Y'/, NO*1/'N'/
      NY=0
C
C          THE ANSWER TO THE QUESTION IS READ
C
10    READ(5,800)YN
800   FORMAT(A)
C
C        NY SET TO 0 IF ANSWER IS NO
C
      IF(YN .EQ. NO)THEN
        NY=0
        GO TO 350
      END IF
C
C        NY SET TO 1 IF ANSWER IS YES
C
      IF(YN .EQ. YES)THEN
        NY=1
        GO TO 350
      END IF
C
C        NY=-1 IF ANSWER IS QUIT
C
      IF(YN.EQ.'Q') THEN
        NY=-1
        GO TO 350
      END IF
C
C          IF Y ,N, OR Q IS NOT ENTERED, YN MUST BE INPUT AGAIN
C
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      CALL BELLL
      WRITE(6,900)
900   FORMAT(' I DO NOT UNDERSTAND !!!',/,
     *' PLEASE ENTER Y (FOR YES), OR N (FOR NO).',/)
      GO TO 10
350   RETURN
      END
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE BELLL
      LOGICAL*1 BEL
      BEL=7
      WRITE(6,100)BEL
100   FORMAT(' ',A1)
      RETURN
      END
C********************
C
C SUBROUTINE H4014
C
C********************
C
C SUBROUTINE H4014 CHANGES THE SIZE OF THE LETTERS PRINTED OUT
C TO THE TEKTRONICS 4014 TERMINAL.
C
C PROGRAMMER: R.A. COX
C
C NOMENCLATURE:
C
C ND      - SMALL LETTERS
C NM      - MEDIUM LETTERS
C NU      - LARGE LETTERS
C NUD     - CONTROL TO DETERMINE WHAT TYPE IS DESERVED
C               0 - SMALL
C               1 - MEDIUM
C               2 - LARGE
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE H4014(NUD)
      DIMENSION NU(2),NM(2),ND(2)
      LOGICAL*1 NU, NM*1, ND*1
C
C LARGE   ESC,9
C
      DATA NU/27,57/
C
C SMALL   ESC,;
C
      DATA ND/27,59/
C
C MEDIUM  ESC,:
C
```

```
      DATA NM/27,58/
      IF(NUD.EQ.2)THEN
      WRITE(6,100)(NU(I),I=1,2)
100   FORMAT(1X,100A1)
      END IF
C
      IF(NUD .EQ. 0)THEN
      WRITE(6,100)(ND(I),I=1,2)
      END IF
C
      IF(NUD .EQ. 1)THEN
      WRITE(6,100)(NM(I),I=1,2)
      END IF
      END
```

A.2.3 Code For Program MODFY

The following pages contain the code developed for the MOD3D subroutine.

```
C
C          ROUTINE MODFY ALLOWS THE USER TO CHOOSE
C       A GEOMETRY MODIFICATION ROUTINE
C
      CHARACTER*4 INPUT
C
C          GET TERMINAL CODE
C
      STATUS=LIB$GET_FOREIGN(INPUT,,N)
      READ(INPUT,900) NTERM
900   FORMAT(I1)
C
C          LIST CHOICES
C
10    WRITE(6,*)' ENTER CODE FOR ROUTINE TO BE SELECTED'
      WRITE(6,*)'    ENTER 0 TO ABORT'
      WRITE(6,*)'  '
      WRITE(6,*)'     1 - POINT BY POINT MODIFICATION'
C
      IF(NTERM.EQ.3) WRITE(6,*)'  '
      READ(5,*,ERR=10) IMETH
      IF(IMETH.LT.0.OR.IMETH.GT.1) GO TO 10
      IMETH=IMETH+1
C
      GO TO (90,20),IMETH
20    CALL MOD3D(NTERM)
      GO TO 100
90    WRITE(6,*)' MODFY ABORT'
100   CONTINUE
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE MOD3D(NTERM)
C
C                 MOD3D IS A SUBROUTINE TO MODIFY THREE DIMENSIONAL
C              GEOMETRY FILES ON A POINT BY POINT BASIS
C
C           PROGRAMMED  23 MARCH 1982       BY M.C. WITTE
C
C        REQUIRED SUBROUTINES
C
C           CHNGE
C           RECNG
C           LIBRARY SUBR
C
C
      DIMENSION COORD(200,3,60),INET(3,60),ICHNG(6000,5)
      CHARACTER*13 NAM,NTIT(60)*60,TITLE*80
     1,POINT*6,OUTPUT*10
      LOGICAL*1 EX
800   FORMAT(I1)
C
C              GET FILE NAME
C
      WRITE(6,*)' ENTER FILE NAME TO BE MODIFIED'
      IF(NTERM.EQ.3) WRITE(6,*)'   '
      READ(5,'(A)') NAM
C
C            READ IN  FILE
C
      INQUIRE(FILE=NAM,EXIST=EX)
      IF(.NOT.EX) THEN
        WRITE(6,*)' FILE ',NAM,' DOES NOT EXIST'
        MP=1
        GO TO 100
      END IF
      OPEN(UNIT=1,FILE=NAM,STATUS='OLD')
      READ(1,965) TITLE
965   FORMAT(A)
      DO K=1,60
        READ(1,955,END=15)INET(3,K),INET(2,K),NTIT(K)
        INET(1,K)=K
        NNET=K
955     FORMAT(2I4,A)
        LEND=INET(2,K)*INET(3,K)
        DO J=1,LEND
           READ(1,960)COORD(J,1,K),COORD(J,2,K),COORD(J,3,K)
960        FORMAT(3F20.5)
        END DO
      END DO
15    CLOSE(UNIT=1)
C
C            RECORD POINT CHANGES
C
      NCHNG=0
20    NCHNG=NCHNG+1
      IF(NTERM.EQ.3)CALL H4014(0)
      WRITE(6,*)' ENTER POINT NUMBERS FOR CHANGES'
      WRITE(6,*)' ENTER "Q" WHEN FINISHED'
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C
C             ENTER POINT 'A' SPECIFICATION
C
21    WRITE(6,*)' ENTER NUMBER FOR POINT A'
      IF(NTERM.EQ.4) WRITE(6,950)
950   FORMAT(/)
      IF(NTERM.EQ.3) WRITE(6,*)'   '
      READ(5,830)POINT
830   FORMAT(A)
      IF(POINT(1:1).NE.'Q') THEN
C
C             DECODE VARIABLE 'POINT'
C
        I=1
        DOWHILE (POINT(1:I).NE.'-')
          I=I+1
          IF(I.GT.3) THEN
            WRITE(6,*)' INVALID POINT SPECIFICATION'
            GO TO 21
          END IF
        END DO
        II=I-1
        III=I+1
        READ(POINT(1:II),840)ICHNG(NCHNG,1)
C
        I=III
        DOWHILE (POINT(1:I).NE.' ')
          I=I+1
          IF(I.GT.6) THEN
            WRITE(6,*)' INVALID POINT SPECIFICATION'
            GO TO 21
          END IF
        END DO
        IF(I.EQ.III) THEN
          WRITE(6,*)' INVALID POINT SPECIFICATION'
          GO TO 21
        END IF
        IE=I-1
        READ(POINT(III:IE),840)ICHNG(NCHNG,2)
C
C               ENTER POINT 'B' SPECIFICATION
C
22      WRITE(6,*)' ENTER NUMBER FOR POINT B'
        IF(NTERM.EQ.4) WRITE(6,950)
        IF(NTERM.EQ.3) WRITE(6,*)'   '
        READ(5,830)POINT
C
C               DECODE VARIABLE 'POINT'
C
        I=1
        DOWHILE (POINT(1:I).NE.'-')
          I=I+1
          IF(I.GT.3) THEN
            WRITE(6,*)' INVALID POINT SPECIFICATION'
            GO TO 22
          END IF
        END DO
```

```
         II=I-1
         III=I+1
         READ(POINT(1:II),840)ICHNG(NCHNG,3)
840      FORMAT(I4)
C
         I=III
         DOWHILE (POINT(1:I).NE.' ')
           I=I+1
           IF(I.GT.6) THEN
             WRITE(6,*)' INVALID POINT SPECIFICATION'
             GO TO 22
           END IF
         END DO
         IF(I.EQ.III) THEN
           WRITE(6,*)' INVALID POINT SPECIFICATION'
           GO TO 22
         END IF
         IE=I-1
         READ(POINT(III:IE),840)ICHNG(NCHNG,4)
C
C                ENTER CHANGE MODE SPECIFICATION
C
23       WRITE(6,*)' ENTER CHANGE CODE'
         IF(NTERM.EQ.4) WRITE(6,950)
         IF(NTERM.EQ.3) WRITE(6,*)'  '
         READ(5,*)ICHNG(NCHNG,5)
         IF(ICHNG(NCHNG,5).NE.1.AND.ICHNG(NCHNG,5).NE.2.AND
     1 .ICHNG(NCHNG,5).NE.3.AND.ICHNG(NCHNG,5).NE.4)THEN
           WRITE(6,*)'      ACCEPTABLE CHANGE CODES ARE'
           WRITE(6,*)'      1 - MOVE POINT B TO POINT A'
           WRITE(6,*)'      2 - MOVE BOTH POINTS TO AN AVERAGE VALUE'
           WRITE(6,*)'      3 - MOVE POINT A TO POINT B'
           WRITE(6,*)'      4 - SPECIFY VALUE FOR BOTH POINTS'
           GO TO 23
         END IF
C
C              ECHO CHANGE FOR CONFIRMATION
C
         WRITE(6,*)' THE PROPOSED CHANGE IS'
         WRITE(6,*)'    POINT A = ',ICHNG(NCHNG,1),'-',ICHNG(NCHNG,2)
         WRITE(6,*)'    POINT B = ',ICHNG(NCHNG,3),'-',ICHNG(NCHNG,4)
         WRITE(6,*)'    CHANGE CODE = ',ICHNG(NCHNG,5)
         WRITE(6,*)' '
         WRITE(6,*)' IS THIS CORRECT?'
         IF(NTERM.EQ.4) WRITE(6,950)
         IF(NTERM.EQ.3) WRITE(6,*)'  '
         CALL YESNO(NY)
         IF(NY.EQ.0) GO TO 21
         GO TO 20
       END IF
       NCHNG=NCHNG-1
C
C              PERFORM CHANGE SEQUENCE
C
       CALL CHNGE(ICHNG,NCHNG,NNET,COORD,INET,NTERM,IERR)
C
C              RECORD GEOMETRY CHANGES
```

```
C
      IF(NCHNG.GT.0) THEN
        WRITE(6,*)' DO YOU WISH TO RECORD GEOMETRY CHANGE'
        IF(NTERM.EQ.4) WRITE(6,950)
        IF(NTERM.EQ.3) WRITE(6,*)'  '
        CALL YESNO(NY)
        IF(NY.EQ.1) THEN
          CALL RECNG(COORD,NNET,INET,NTERM,NAM,TITLE,NTIT)
        END IF
      END IF
C
100   WRITE(OUTPUT,940) MP
940   FORMAT('$@TRANSF ',I1)
      RETURN
      END
```

```
      SUBROUTINE CHANGE(ICHNG,NCHNG,NNET,COORD,INET,NTERM,IERR)
C
C              MODIFY ONE OR A GROUP OF POINTS IN A THREE
C            DIMENSIONAL MODEL
C
C         PROGRAMMED  3 MARCH,1982  BY M.C. WITTE
C
C         FOR USE WITH PROGRAM MOD3D
C
      DIMENSION ICHNG(6000,5),COORD(200,3,60),INET(3,60)
      IERR=0
C
C           LOOP THROUGH NCHNG CHANGES
C
      DO I=1,NCHNG
C
C           FIND SPECIFIED NETWORK FOR POINT A
C
        JA=1
        DOWHILE (ICHNG(I,1).NE.INET(1,JA))
          JA=JA+1
C
C           NETWORK NOT FOUND
C
          IF(JA.GT.NNET) THEN
            WRITE(6,*)' POINT CHANGE ERROR ON CHANGE NO. ',I
            WRITE(6,*)' NETWORK ',ICHNG(I,1),' NOT FOUND'
            WRITE(6,*)' CHANGE ABORTED'
            IERR=1
            GO TO 100
          END IF
        END DO
C
C           FIND POSITION OF POINT A IN COORD
C
        NSKIPA=0
        LEND=JA-1
        DO K=1,LEND
          NSKIPA=NSKIPA+INET(2,K)*INET(3,K)
        END DO
C
C           POINT NOT FOUND
C
        IF(ICHNG(I,2).GT.INET(2,JA)*INET(3,JA)) THEN
          WRITE(6,*)' POINT CHANGE ERROR ON CHANGE NO. ',I
          WRITE(6,*)' POINT INDEX OUT OF BOUNDS'
          WRITE(6,*)' CHANGE ABORTED'
          IERR=1
          GO TO 100
        END IF
        NSKIPA=NSKIPA+ICHNG(I,2)
C
C           FIND SPECIFIED NETWORK FOR POINT B
C
        JB=1
        DOWHILE (ICHNG(I,3).NE.INET(1,JB))
          JB=JB+1
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C
C           NETWORK NOT FOUND
C
          IF(JB.GT.NNET) THEN
            WRITE(6,*)' POINT CHANGE ERROR ON CHANGE NO. ',I
            WRITE(6,*)' NETWORK ',ICHNG(I,3),' NOT FOUND'
            WRITE(6,*)' CHANGE  ABORTED'
            IERR=1
            GO TO 100
          END IF
        END DO
C
C           FIND POSITION OF POINT B IN COORD
C
        NSKIPB=0
        LEND=JB-1
        DO K=1,LEND
          NSKIPB=NSKIPB+INET(2,K)*INET(3,K)
        END DO
C
C           POINT NOT FOUND
C
        IF(ICHNG(I,4).GT.INET(2,JB)*INET(3,JB)) THEN
          WRITE(6,*)' POINT CHANGE ERROR ON CHANGE NO. ',I
          WRITE(6,*)' POINT INDEX OUT OF BOUNDS'
          WRITE(6,*)' CHANGE ABORTED'
          IERR=1
          GO TO 100
        END IF
        NSKIPB=NSKIPB+ICHNG(I,4)
C
C          PERFORM CHANGE ACCORDING TO CHANGE MODE
C           SPECIFICATION
C
C
        GO TO (20,40,60,80),ICHNG(I,5)
C
        WRITE(6,*)' INVALID CHANGE MODE SPECIFICATION'
        WRITE(6,*)' CHANGE NO. ',I
        WRITE(6,*)' CHANGE ABORTED'
        IERR=1
        GO TO 100
C
C            MOVE POINT B TO POINT A
C
20      DO L=1,3
          COORD(NSKIPB,L,JB)=COORD(NSKIPA,L,JA)
        END DO
        GO TO 100
C
C            MOVE BOTH POINTS TO AN AVERAGE VALUE
C
40      XAVE=(COORD(NSKIPA,1,JA)+COORD(NSKIPB,1,JB))/2.
        YAVE=(COORD(NSKIPA,2,JA)+COORD(NSKIPB,2,JB))/2.
        ZAVE=(COORD(NSKIPA,3,JA)+COORD(NSKIPB,3,JB))/2.
C
        COORD(NSKIPA,1,JA)=XAVE
        COORD(NSKIPB,1,JB)=XAVE
```

```
          COORD(NSKIPA,2,JA)=YAVE
          COORD(NSKIPB,2,JB)=YAVE
          COORD(NSKIPA,3,JA)=ZAVE
          COORD(NSKIPB,3,JB)=ZAVE
          GO TO 100
C
C              MOVE POINT A TO POINT B
C
60        DO L=1,3
            COORD(NSKIPA,L,JA)=COORD(NSKIPB,L,JB)
          END DO
          GO TO 100
C
C              SPECIFY VALUE COMMON TO BOTH POINTS
C
80        WRITE(6,*)'ENTER X,Y,Z VALUE FOR CHANGE NO. ',I
          IF(NTERM.EQ.4) WRITE(6,950)
          IF(NTERM.EQ.3) WRITE(6,*)'  '
950       FORMAT(/)
          READ(5,*,ERR=80) XAVE,YAVE,ZAVE
          COORD(NSKIPA,1,JA)=XAVE
          COORD(NSKIPB,1,JB)=XAVE
          COORD(NSKIPA,2,JA)=YAVE
          COORD(NSKIPB,2,JB)=YAVE
          COORD(NSKIPA,3,JA)=ZAVE
          COORD(NSKIPB,3,JB)=ZAVE
100       CONTINUE
        END DO
        RETURN
        END
FORTRAN STOP
$
```

```
      SUBROUTINE RECNG(COORD,NNET,INET,NTERM,INAM,TITLE,NTIT)
C
C                RECORD 3-D GEOMETRY CHANGES IN A FILE
C
C          PROGRAMMED  23 MARCH 1982       M.C. WITTE
C
C           FOR USE WITH PROGRAM MOD3D
C
      DIMENSION COORD(200,3,60),INET(3,60)
      CHARACTER*3 COM,INAM*13,NWNAM*13,NTIT(60)*60,IDUM*2
     1,TITLE*80
C
      WRITE(6,*)' REPLACE FILE OR CREATE NEW FILE,(REP,NEW)'
      IF(NTERM.EQ.4) WRITE(6,950)
      IF(NTERM.EQ.3) WRITE(6,*)'  '
950   FORMAT(/)
      READ(5,900) COM
900   FORMAT(A)
      IF(COM.EQ.'NEW') THEN
        WRITE(6,*)' ENTER NEW FILE NAME (13 CHARS WITH EXTENSION)'
        IF(NTERM.EQ.4) WRITE(6,950)
        IF(NTERM.EQ.3) WRITE(6,*)'  '
        READ(5,900) NWNAM
      END IF
C
C             OPEN ORIGINAL FILE
C
      IF(COM.EQ.'REP') THEN
        OPEN(UNIT=1,FILE=INAM,STATUS='OLD')
      END IF
C
C             OPEN NEW FILE
C
      IF(COM.EQ.'NEW') THEN
        OPEN(UNIT=2,FILE=NWNAM,STATUS='NEW')
      ELSE
        OPEN(UNIT=2,FILE=INAM,STATUS='NEW')
      END IF
C
C            WRITE INTO NEW FILE
C
C
C            WRITE FILE TITLE
C
      WRITE(2,900)TITLE
C
C             WRITE NETWORK DATA
C
      DO K=1,NNET
910     FORMAT(I4,I4,A)
        WRITE(2,910)INET(3,K),INET(2,K),NTIT(K)
        LEND=INET(3,K)*INET(2,K)
        DO J=1,LEND
          WRITE(2,920)COORD(J,1,K),COORD(J,2,K),COORD(J,3,K)
920       FORMAT(3F20.5)
        END DO
      END DO
```

```
100    CLOSE(UNIT=2)
       IF(COM.EQ.'REP') THEN
         CLOSE(UNIT=1,DISP='DELETE')
       END IF
C
       RETURN
       END

FORTRAN STOP
$
```

A.2.4 Code For Program CREFIL

The following pages contain the code developed for the program CFIFIL.

ORIGINAL PAGE IS
OF POOR QUALITY

```
C
C           CREFIL CREATES A PRELIMINARY GEOMETRY FILE
C           FOR INPUT TO THE SMITH-HESS GEOMETRY PANELING
C           PROCEDEURE
C
      CHARACTER*4 INPUT
C
C           GET NTERM
C
      STATUS=LIB$GET_FOREIGN(INPUT,,N)
      READ(INPUT,900) NTERM
900   FORMAT(I1)
      CALL WRTCON(NTERM)
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE WRTCON(NTERM)
C
C
C               THIS ROUITNE ALLOWS THE USER TO SELECT THE METHOD
C           BY WHICH HE WISHES TO CONSTRUCT A PRELIMINARY
C           GEOMETRY FILE
C
C
10    WRITE(6,*)' ENTER THE CODE CORRESPONDING TO THE METHOD'
      WRITE(6,*)' THAT YOU DESIRE'
      WRITE(6,*)' ENTER THE NEGATIVE OF THE CODE TO CONTINUE'
      WRITE(6,*)' A PREVIOUS GEOMETRY DEFINITION'
      WRITE(6,*)'  '
      WRITE(6,*)'          0 - ABORT '
      WRITE(6,*)'          1 - POINT BY POINT DATA ENTRY'
      WRITE(6,*)'          2   INTERACTIVE CONFIGURATION '
      WRITE(6,*)'                          DEVELOPMENT'
      WRITE(6,*)'  '
C
C               READ METHOD CODE
C
      IF(NTERM.EQ.3) WRITE(6,*)'   '
      IF(NTERM.EQ.4) WRITE(6,950)
950   FORMAT(/)
      READ(5,*,ERR=10)ICOD
      IF(ABS(ICOD).GT.2) GO TO 10
      IF(ICOD.EQ.0) GO TO 100
C
C               EXECUTE DESIRED METHOD
C
      GO TO (40,80),ABS(ICOD)
      GO TO 10
C
C              INPUT GEOMETRY POINT BY POINT
C
40    CALL PTBYPT(NTERM,ICOD)
      RETURN
C
C              INTERACTIVELY CONSTRUCT THE CONFIGURATION
C
80    CALL  INCON(NTERM,ICOD)
      RETURN
C
C              ABORT GEOMETRY DEFINITION
C
100   RETURN
      END
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
      SUBROUTINE PTBYPT(NTERM,ICOD)
C
```

```
C               GENERATE CONFIGURATION GEOMETRY FILE BY
C           ACCEPTING POINT BY POINT INPUT FROM USER
C
C
      CHARACTER*13 PFNAME,NETNAM(61)*60,TITLE*80
      DIMENSION COORD(400,3,60),INET(2,60),ISTAT(400,2,60),NWAKE(61)
      LOGICAL*1 EX
      NNET=0
C
C            ENTER PRELIMINARY GEOMETRY FILE NAME
C
10    WRITE(6,*)' ENTER PRELIMINARY GEOMETRY FILE NAME (13 CHARS)'
      IF(NTERM.EQ.3) WRITE(6,*)'   '
      IF(NTERM.EQ.4) WRITE(6,950)
950   FORMAT(/)
      READ(5,965,ERR=10) PFNAME
C
C            IF ADDING TO AN EXISTING FILE
C
      IF(ICOD.LT.0) THEN
        INQUIRE(FILE=PFNAME,EXIST=EX)
        IF(.NOT.EX) THEN
          WRITE(6,*)' FILE DOES NOT EXIST'
          GO TO 200
        END IF
C
C            READ FROM EXISTING FILE
C
        OPEN(UNIT=1,FILE=PFNAME,STATUS='OLD')
        READ(1,965) TITLE
965     FORMAT(A)
        DO K=1,60
      READ(1,955,END=15)INET(2,K),INET(1,K),NETNAM(K)
          NNET=K
955       FORMAT(2I4,A)
          LEND=INET(1,K)*INET(2,K)
          DO J=1,LEND
      READ(1,960)COORD(J,1,K),COORD(J,2,K),COORD(J,3,K),
     2 ISTAT(J,1,K),ISTAT(J,2,K)
960          FORMAT(3F20.5,1X,2I1)
          END DO
        END DO
15      CLOSE(UNIT=1,DISP='DELETE')
      ELSE
C
C            ENTER A FILE TITLE
C
20      WRITE(6,*)' ENTER A DESCRIPTIVE FILE TITLE (80 CHARS)'
        IF(NTERM.EQ.3) WRITE(6,*)'   '
        IF(NTERM.EQ.4) WRITE(6,950)
        READ(5,965,ERR=20) TITLE
      END IF
C
C            WRITE USER INSTRUCTIONS
C
      WRITE(6,*)'         Begin entering network data.'
      WRITE(6,*)'         Enter "QUIT" when finished,or "ABORT" to'
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      WRITE(6,*)'        to terminate procedeure'
      WRITE(6,*)'   '
      WRITe(6,*)'      enter -999.,-999.,-999. to correct the imediately'
      WRITE(6,*)'        previuos entry'
C
C     WRITE(6,*)' DO YOU WISH ADDITIONAL INFORMATION ?'
C     IF(NTERM.EQ.3) WRITE(6,*)'   '
C     IF(NTERM.EQ.4) WRITE(6,950)
C     CALL YESNO(NY)
C     IF(NY.EQ.1) THEN
C       WRITE(6,*)
C       WRITE(6,*)
C       WRITE(6,*)
C       WRITE(6,*)
C       WRITE(6,*)
C       WRITE(6,*)
C       WRITE(6,*)
C       WRITE(6,*)
C       WRITE(6,*)
C       WRITE(6,*)
C     END IF
C
C            ENTER NETWORK TITLE
C
40    WRITE(6,*)' ENTER NETWORK TITLE FOR NETWORK NO. ',NNET+1
      IF(NTERM.EQ.3) WRITE(6,*)'   '
      IF(NTERM.EQ.4) WRITE(6,950)
      READ(5,965,ERR=40) NETNAM(NNET+1)
      IF(NETNAM(NNET+1)(1:1).EQ.'A') GO TO 200
      IF(NETNAM(NNET+1)(1:1).EQ.'Q') GO TO 100
      IF(NNET.EQ.60) THEN
        WRITE(6,*)' MAXIMUM NETWORK COUNT EXCEEDED'
        GO TO 100
      END IF
C
C          ENTER INET
C
50    WRITE(6,*)' ENTER NUMBER OF POINTS PER COLUMN'
      WRITE(6,*)'        FOR NETWORK ',NETNAM(NNET+1)
      IF(NTERM.EQ.3) WRITE(6,*)'   '
      IF(NTERM.EQ.4) WRITE(6,950)
      READ(5,*,ERR=50) INET(2,NNET+1)
C
55    WRITE(6,*)' ENTER NUMBER OF COLUMNS'
      WRITE(6,*)'        FOR NETWORK ',NETNAM(NNET+1)
      IF(NTERM.EQ.3) WRITE(6,*)'   '
      IF(NTERM.EQ.4) WRITE(6,950)
      READ(5,*,ERR=55) INET(1,NNET+1)
      WRITE(6,*) ' ENTER 0 IF NETWORK ',NETNAM(NNET+1),'   '
      WRITE(6,*) ' IS A NONLIFTING COMPONENT.'
      WRITE(6,*) '   '
      WRITE(6,*) ' ENTER 1 IF NETWORK ',NETNAM(NNET+1),'   '
      WRITE(6,*)' IS A LIFTING COMPONENT.'
      WRITE(6,*) ' ENTER 0 OR 1 ? '
      READ(5,*) JSTAT
      WRITE(6,*) '   '
      INETCK=-10.0
```

```
      IF(JSTAT.EQ.1) THEN
      WRITE(6,*) ' DO YOU WISH TO DEFINE A WAKE '
      WRITE(6,*) ' FOR THIS COMPONENT ? '
      WRITE(6,*) ' ENTER YES OR NO . '
      INETCK=-10.0
      CALL YESNO(NY)
      IF(NY.EQ.1) THEN
58    WRITE(6,*) ' ENTER THE NUMBER OF WAKE POINTS TO BE DEFINED'
      WRITE(6,*) ' FOR EACH COLUMN . '
      READ(5,*,ERR=58) NWAKE(NNET+1)
      INETCK=INET(2,NNET+1)
      INET(2,NNET+1)=INET(2,NNET+1)+NWAKE(NNET+1)
      END IF
      END IF
C
C          ENTER X,Y,Z COORDINATES
C
      NC=INET(1,NNET+1)
      NP=INET(2,NNET+1)
      IF(NC*NP.GT.400) THEN
        WRITE(6,*)' MAXIMUM NUMBER OF POINTS IS 400'
        GO TO 50
      END IF
      M=1
      I=1
      DOWHILE (I.LE.NC)
        J=1
      ISTAT(M,2,NNET+1)=JSTAT
      IF(J.EQ.1) ISTAT(M,1,NNET+1)=1
      IF(M.EQ.1) ISTAT(M,1,NNET+1)=2
60      WRITE(6,*)' FOR COLUMN NO. ',I,' ,NETWORK ',NETNAM(NNET+1)
        IF(NTERM.EQ.3) WRITE(6,*)'  '
        IF(NTERM.EQ.4) WRITE(6,950)
        DOWHILE (J.LE.NP)
      ISTAT(M,2,NNET+1)=JSTAT
      IF(J.GT.1) ISTAT(M,1,NNET+1)=0
      IF(J.EQ.INETCK+1) ISTAT(M,1,NNET+1)=4
C     IF(I.EQ.NC.AND.J.EQ.NP) ISTAT(M,1,NNET+1)=3
65        WRITE(6,*)' ENTER X,Y,Z - POINT NO. ',J
          IF(NTERM.EQ.3) WRITE(6,*)'  '
          IF(NTERM.EQ.4) WRITE(6,950)
          READ(5,*,ERR=65)(COORD(M,K,NNET+1),K=1,3)
          IF(ABS(COORD(M,1,NNET+1)+999.).LT..0001) THEN
            IF(J.EQ.1) THEN
              J=NP
              M=M-1
              I=I-1
              GO TO 60
            ELSE
              J=J-1
              M=M-1
              GO TO 65
            END IF
          END IF
          M=M+1
          J=J+1
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
          END DO
          I=I+1
        END DO
C       WRITE(6,*) ' IS THIS LAST NETWORK THAT YOU WISH '
C       WRITE(6,*) ' TO INPUT ?    ( ENTER YES OR NO ) .'
C       CALL YESNO(NY)
C       IF(NY.EQ.1) THEN
C       ISTAT(M-1,1,NNET+1)=3
C       END IF
        NNET=NNET+1
        GO TO 40
C
C              WRITE INFO TO FILE
C
100     OPEN(UNIT=1,FILE=PFNAME,STATUS='NEW')
        ISTAT(M-1,1,NNET)=3
C
C              WRITE FILE TITLE
C
        WRITE(1,965)TITLE
C
C               WRITE NETWORK DATA
C
        DO K=1,NNET
          WRITE(1,955)INET(2,K),INET(1,K),NETNAM(K)
          LEND=INET(2,K)*INET(1,K)
          DO J=1,LEND
        WRITE(1,960)COORD(J,1,K),COORD(J,2,K),COORD(J,3,K),ISTAT(J,1,K),
       2 ISTAT(J,2,K)
          END DO
        END DO
        CLOSE(UNIT=1)
C
200     CONTINUE
        RETURN
        END
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
        SUBROUTINE INCON(NTERM,ICOD)
C
C               GENERATE CONFIGURATION GEOMETRY FILE BY
C           INTERACTING WITH THE USER
C
        WRITE(6,*)' OPTION NOT AVAILABLE AT THIS TIME'
        RETURN
        END
```

APPENDIX B

USERS' GUIDE TO THE ENGINE SIMULATION MODULE

B.1 Program Instructions

The Engine Simulation Module was developed to provide the user a means by which to include the aerodynamic/propulsive interactions in a potential flow solution of a STOL configuration. It accomplishes this by adding networks and boundary conditions to existing PAN AIR files to obtain plume simulation. Currently, the code is capable of handling a supersonic exhaust flow that is ejected at angles ranging from 0 deg to approximately 30 deg to the freestream flow. For convenience, the Engine Simulation Module is stored in a file named "ADDJET". It is resident on the NASA Ames' VAX 11/780 computer system and requires that an input file and a PAN AIR file be in the same subdirectory as the executable version of the program. The input file, which must be prepared by the user, is named 'ADDJET.INP'. Information in this file specifies where the plume is to be placed on the panelling arrangement and properties of the exhaust flow at the nozzle exit. The ADDJET.INP file must be prepared according to the following instructions.

Record 1: TITLE

Up to 50 alphanumeric characters may be used to define a problem title, which will appear in the output.

Record 2: DUMP INDICATORS

This card should normally be left blank or contain zeros. It may be used, however, to obtain a dump of the internal variables from selected subroutines by inserting a "1" in the appropriate column as indicated below:

Col.	Subroutine
1	ADDJET

2	READNET
3	CENTER
4	CLINE
5	START
6	EXTEND
7	PBASE
8	WAKE

The user must refer to the fortran listing for information regarding the definition of the variables in the subroutine dumps.

Record 3. PLUME LOCATION

The plume-location information is entered in a namelist input format, which allows variables to be specified with one variable per line or several variables per line. In either case, the variables must be separated by commas. The following parameters must be specified in Record 3.

Record 3A $PADATN

This is the name of the upcoming data set and must be input exactly as shown. It is short for PAN AIR Data Namelist. The "$" character must not be placed in column 1.

Record 3B OLDFIL

This is the file name of the original PAN AIR file to which the plume is to be added. The file name must be enclosed within single quotes, with no leading or trailing blanks. Example:

OLDFIL = 'FIGHTER.JOB'

Record 3C NEWFIL

The original PAN AIR file (OLDFIL) is not destroyed or altered by the Engine Simulation Module. Instead, a completely new file is created which contains the

basic panelling arrangement with the plume networks added. The user must specify the name of the new file through this parameter. Example:

NEWFIL = 'FIGHTER.NEW'

Record 3D ENAME

This is the name of the network that forms the nozzle exit in the original PAN AIR file. It must correspond exactly with the name used in Record N2a of the PAN AIR input and must be a single network that closely approximates the exit area of the nozzle. The input data for this network will appear in the modified file but will be automatically nullified by the ADDJET code, which places a "*/" in columns 1 and 2 of each line of data for that network. Example:

ENAME = 'NOZZLE-EXIT'

Record 3E ABUT (I)

These parameters specify the networks in the original PAN AIR file that will abut with the plume. A maximum of 6 networks may be specified.

ABUT (1) = 'NOZZLE-UPPER'
ABUT (2) = 'NOZZLE-LOWER'
ABUT (3) = 'NOZZLE-SIDE'

Record 3F IEDGE(I)

These terms specify which edges of the networks in Record 3E form the abutment with the plume. The values are integer numbers 1, 2, 3 or 4 and correspond to the PAN AIR nomenclature. Records 3E and 3F must have the same number of input points. Record 3F must be terminated by including a "$" character after the last input value. Example:

IEDGE (1) = 2
IEDGE (2) = 1
IEDGE (3) = 4 $

Record 4. PLUME PROPERTIES

The plume property information is also a namelist input format. It must be started and terminated with the "$" character and variables must be separated by commas. The following parameters must be specified in Record 4.

Record 4A $RAXINN

This is the name of the upcoming data set and must be input exactly as shown. It is short for RAXJET Input Namelist. The "$" character must not be placed in column 1.

Record 4B XNPR

Ratio of jet total pressure to freestream static pressure (nozzle pressure ratio). Example:

EXPR = 2.5

Record 4C EMJET

Nozzle exit Mach number. The RAXJET code is limited to supersonic jets (EMJET ≥ 1.05). Example:

EMJET = 2.00

Record 4D TTJET

Nozzle exit total temperature in degrees Rankin (deg-R). Example:

TTJET = 530.0

Record 4E THLIP

Nozzle exit lip angle in degrees, (i.e. the initial angle between the plume boundary and the freestream direction). Example:

THLIP = 4.5

Record 4F THETAJ

The injection angle of the exhaust plume relative to the freestream direction. Example:

THETAJ = 15.0

Record 4G AMINF

Ambient Mach number at infinity. The RAXJET code is restricted to subsonic external flow (.1 AMINF .8) Example:

AMINF = .20

Record 4H JDUM

This term is normally set to zero or it may be omitted from the input stream. If JDUM = 1, the ADDJET program will not call the RAXJET subroutines to add the plume, but instead, will add a solid, cylindrical plume. Since the RAXJET code requires considerable time to execute, this option allows a run to be made in a few seconds of CPU time to verify that all input cards are being correctly processed before executing the full code. Example:

JDUM = 1

Record 4I IPRINT

The output format for the unmodified RAXJET code, which is discussed in Reference 32, is obtainable from the ADDJET code along with one additional page of parameters computed for the inviscid plume. A user should not normally need this information, but there are several options available for handling this output.

IPRINT = 0 Data not saved
IPRINT = 1 Data saved in file "RAXJET.OUT"

IPRINT = 2 Data printed, then file is deleted

IPRINT = 3 Data printed and also saved in file "RAXJET.OUT"

Since this is the last input parameter in Record 4, it must be followed with a "$" to terminate the RAXINN data set. Example:

IPRINT = 1 $

The ADDJET code is easy to run and highly automated; however, there still remains a few items that the user must be aware of and manually correct if necessary. Several items to check are outlined below.

- o The plume and the plume wake networks generated by the ADDJET code are panelled irrespective of any networks that may be near or abut with the plume. Therefore, the user may need to manually define the abutment between the wing wake, for instance, and the plume and its wake.

- o Wake networks that may have been originally included in a panelling arrangement to form the closure of the nozzle must be manually removed.

- o The exit network is removed from the input data stream. Any specified abutments involving this network should be removed from the abutment specifications list.

- o Since the exit network is removed, the ordering index is changed for all of the original PAN AIR networks that follow it. Therefore, the user must review the abutment specification list and the network-images cards (PAN AIR records GE2, SF2, and FM8) and make any changes that may be required).

B.2 Sample Problem

The isolated nacelle model discussed in Section 5.5 is employed again in this appendix in a sample problem. Input data for this model are presented in this section, followed by a listing of the new PAN AIR file that was created by the program. The input file is described in the User's Guide to the Engine Simulation Module (Appendix B.1), and some comments regarding the output file are presented in this section.

The job control language (JCL) required to submit a PAN AIR file to the CRAY computer was contained within the same file as the original PAN AIR input. These control cards are automatically transferred to the output file as shown, along with all comment cards preceeding the first readable card of the original PAN AIR file. A banner is placed immediately ahead of the "BEGIN GLOBAL DATA" card of the output file to direct attention to the fact that plume networks have been added. Within this banner, the parameters specified in the input file are printed so the user can identify the source file of the original PAN AIR data and also the properties of the simulated exhaust plume. Note the characters "/ADDJET" that are placed to the right-hand side of the banner. This designation is placed on every line that is entered or modified by the ADDJET code so that all file alterations can be easily identified.

After the initial banner, the next occurence of a modified card in this sample problem is at the nozzle exit (NETWORK = EXIT - 4). Although these cards are nullified by the ADDJET code, they are included in the data stream for documentation. No cards are ever removed by the ADDJET code.

The plume segments follow the last network in the original PAN AIR file. This input will consist of typical coordinate data to define the panel corner points and a set of velocities to represent entrainment. It is noted that entrainment is indicated by a negative velocity normal to the panel center points. The last two sets of input cards generated by the ADDJET code are for the plume base and for the wake that emanates from the base of the plume. All cards in and below the Geometric Edge Matching section are copied to the new file without any changes.

B.3 Program Description and Listings

The structure of the Engine Simulation Module, Figure B-1, illustrates the relationship between the various subroutines in the program. The function of each subroutine is briefly discussed in the following statements.

ADDJET -

This is the master control for the Engine Simulation Module. It reads the data from the RAXJET.INP file, opens the necessary files for output, and calls the appropriate subroutines to execute the remainder of the code.

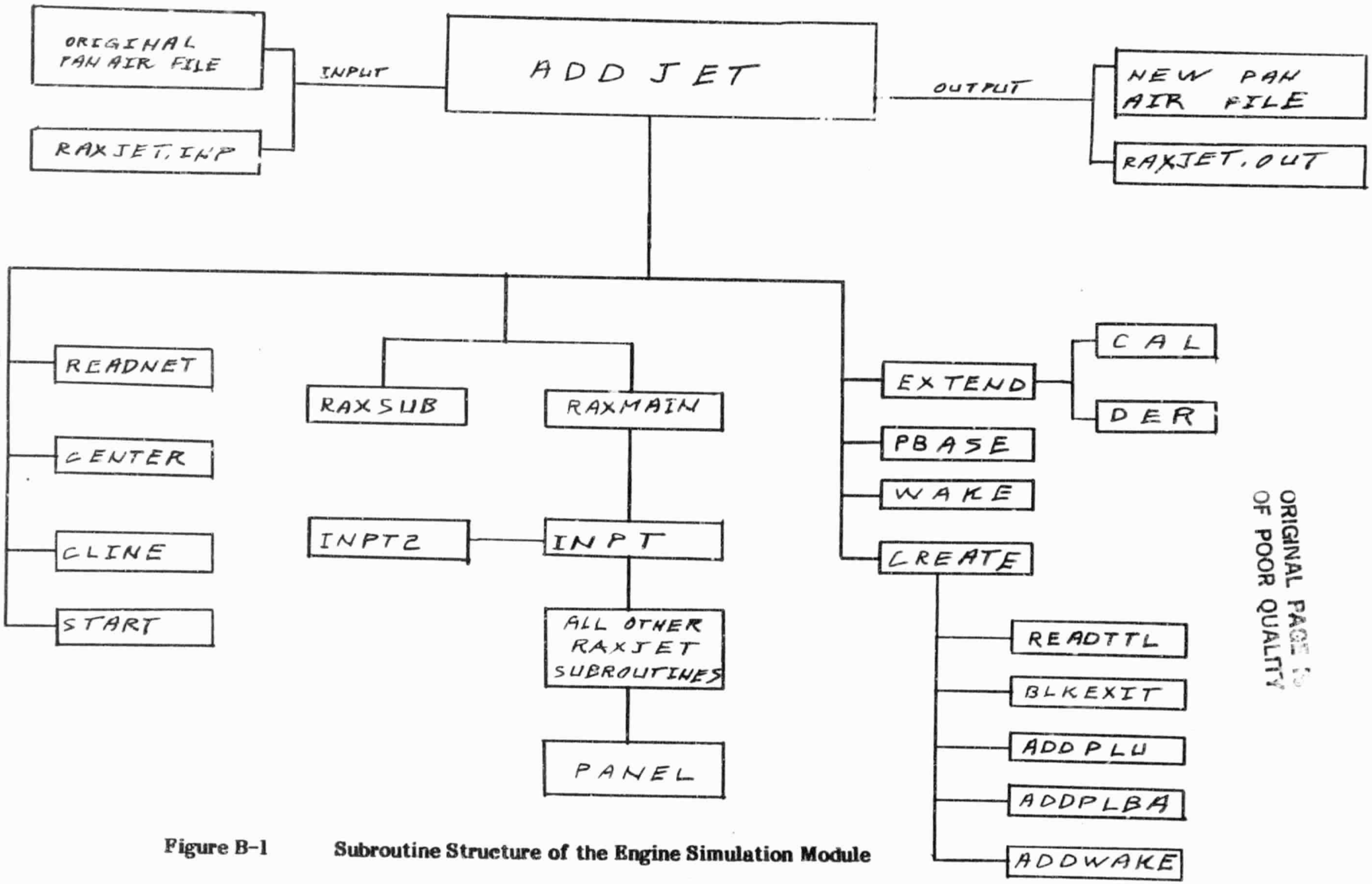


Figure B-1 Subroutine Structure of the Engine Simulation Module

READNET -

This subroutine starts at the current position of the PAN AIR file and reads records until all the information for the next network is obtained. The task of this subroutine is complicated by the fact that PAN AIR input is unformatted and is normally read by a data management system that is not available on the NASA VAX system.

CENTER -

This subroutine is called immediately after the nozzle exit network data are read by subroutine READNET. It uses these data to compute the geometric center of the plume at the X-station where it abuts the nozzle exit. It also checks to determine if the plume network will abut either or both of the planes of symmetry at Y = 0 and Z = 0.

CLINE -

This subroutine computes the coefficients of a second-order equation that defines the path of the plume centerline. The initial jet injection angle is accounted for.

START -

This subroutine defines the coordinates for the edge of the plume that abuts the solid networks. It accomplishes this by determining the coordinates of the solid network edges that abut the plume (as specified in the input) and defining the plume edge for exact corner point matching.

RAXSUB

This subroutine allows the ADDJET code to be executed without calling the RAXJET subroutines. It returns the characteristics for a solid, cylindrical plume and allows the remainder of the code to run with minimal CPU time.

RAXMAIN -

This subroutine was originally the controlling subroutine for the unmodified RAXJET code. It has been modified for use in the ADDJET code by adding (1) common blocks to transfer variables between the ADDJET code and the RAXJET code, and (2) statements required to open and close the various output and scratch files.

INPT -

This subroutine was also part of the original RAXJET code. It has been modified for use in this code by changing the source from which it requests the user supplied input

variables. Rather than read an input file, it obtains the input values from the file RAXJET.INP data or from subroutine INPT2.

INPT2 -

This subroutine is called by INPT. It presets many of the parameters that are normally specified by a user to values that are proper for a low-speed analysis, thus relieving the user of the chore of making these selections.

PANEL -

Subroutine PANEL was written to convert the RAXJET-computed variables to a form that could be used in the remaining subroutines of the ADDJET code. The primary function is the integration of radii of the equivalent and inviscid plume shapes to obtain inflow velocities normal to the inviscid plume boundary.

EXTEND -

This subroutine receives the plume coordinates along the abutment with the solid networks, the plume centerline path, the plume expansion characteristics, and entrainment factors. From this information, it computes the coordinates of the panel corner points for a 3-dimensional plume and the inflow velocities at the panel center points.

CAL,DER -

These subroutines perform a spline curvefit that is used in interpolating between data points in subroutine EXTEND.

PBASE -

This subroutine computes the data for the plume base.

WAKE -

This subroutine computes the data for the plume wake.

CREATE -

This subroutine controls the creation of the new PAN AIR file. It reads the old file, determines where the new or modified records should be, and calls the appropriate subroutines to make the actual insertion.

READTTL -

This subroutine is used by CREATE to determine the title of the upcoming network in the original PAN AIR file.

BLKEXIT -

Subroutine BLKEXIT is called by CREATE to nullify the cards in the network defining the original nozzle exit.

ADDPLU -

This subroutine is called by CREATE to add records to the new PAN AIR file for the plume networks.

ADDPLBA -

This subroutine is called by CREATE to add records to the new PAN AIR file for the plume base networks.

ADDWAKE -

This subroutine is called by CREATE to add the wakes that emanate from the plume base.

SAMPLE PROBLEM INPUT

```
   ISOLATED NACELLE PROBLEM (NPR= 8.47)
111111111111111
 $PADATN
      OLDFIL = 'NACELLE.JOB',
      NEWFIL = 'NACELLE.PLU',
      ENAME  = 'EXIT-4',
       ABUT(1) = 'NOZZLE-3',
      IEDGE(1)= 2       $
 $RAXINN
       XNPR = 8.47,
       EMJET = 1.31,
       TTJET = 560.,
       THLIP = 2.05,
       THETAJ = 0.0,
       AMINF = 0.6,
       JDUM = 0,
       IPRINT = 1        $
```

SAMPLE PROBLEM OUTPUT FILE

```
PANAIR,MICR.
USER,FAEGDH,YSAW.
JOB,JN=GENDYN,T=300,US=FAEGDH.NA=HOWELL,PANAIR-RUN.
ACCOUNT,AC=G3293F0.
ACCESS,DN=$PROC,PDN=REMOTEP,ID=AMESLIB.
REMOTE,FAEGDH,YSAW,G3293F0,DN=$OUT,DEFER,IDT=STAICB.
RELEASE,DN=$PROC.
ACCESS(DN=TEMP,PDN=PAPROCS,ID=PANAIR)
COPYD(I=TEMP,O=$PROC)
RELEASE(DN=TEMP)
REWIND(DN=$PROC)
FINDPF(A=2SW,ID=FAEGDH,MECIN,PPP,TEMP)
REWIND(DN=FT10)
DISPOSE(DN=FT10)
AUDIT.
**EOR**
*/
*/
*/
*/
*/
*/
*/
*/
*/
*/
*/ DIP INPUT - FILE NACELLE.JOB
*/
*/
*/          ISOLATED NACELLE MODEL
*/
*/
*/                    5 ADDITIONAL CUTS HAVE BEEN ADDED NEAR THE
*/                    KNEE OF THE NOZZLE
*/
*/
*/
*/                   THIS INPUT REPRESENTS ONLY ONE FOURTH OF THE
*/                    COMPLETE CONFIGURATION.
*/
*/
*/
*/          BASED ON NASA  TP 1766 ::CONFIGURATION D-1.22-L
*/
*/             5-NETWORKS
*/
*/            1)FORWARD CONIC SECTION --------    6 PANELS
*/            2)CYLINDRICAL MID-SECTION ------   27 PANELS
*/            3)BOATTAIL ---------------------   78 PANELS
*/            4)PLUME ------------------------  126 PANELS
*/            5)PLUME BASE -------------------    6 PANELS
*/            6)CYLINDRICAL WAKE -------------    6 PANELS
*/                                              ----------
*/                                     TOTAL:  249 PANELS
*/
*/
*/
```

```
*/
*/   *********************************************************************** /ADDJET
*/   *********************************************************************** /ADDJET
*/   ***                                                                 *** /ADDJET
*/   ***           THE PLUME SIMULATION IN THIS PAN AIR                  *** /ADDJET
*/   ***              FILE WAS INCORPORATED WITH THE                     *** /ADDJET
*/   ***                                                                 *** /ADDJET
*/   ***                A D D J E T     C O D E                          *** /ADDJET
*/   ***                                                                 *** /ADDJET
*/   ***                      WRITTEN BY                                 *** /ADDJET
*/   ***        GEORGE A. HOWELL AND ISHWAR C. BHATELEY                  *** /ADDJET
*/   ***              GENERAL DYNAMICS CORPORATION                       *** /ADDJET
*/   ***                  FORT WORTH DIVISION                            *** /ADDJET
*/   ***                   FORT WORTH, TEXAS                             *** /ADDJET
*/   ***                 PHONE (817) 763-1161                            *** /ADDJET
*/   ***                                                                 *** /ADDJET
*/   ***                                                                 *** /ADDJET
*/   ***       THE 2-D JET CHARACTERISTICS WERE COMPUTED                 *** /ADDJET
*/   ***     WITH THE RAXJET PLUME MODELING CODE DEVELOPED               *** /ADDJET
*/   ***                BY RICHARD G. WILMOTH                            *** /ADDJET
*/   ***             NASA/LANGLEY RESEARCH CENTER                        *** /ADDJET
*/   ***                                                                 *** /ADDJET
*/   ***                                                                 *** /ADDJET
*/   ***   ISOLATED NACELLE PROBLEM (NPR= 8.47)                          *** /ADDJET
*/   ***                                                                 *** /ADDJET
*/   ***         ORIGINAL FILE              =  NACELLE.JOB               *** /ADDJET
*/   ***         NEW FILE                   =  NACELLE.PLU               *** /ADDJET
*/   ***                                                                 *** /ADDJET
*/   ***         NOZZLE PRESSURE RATIO      = 8.470                      *** /ADDJET
*/   ***         JET EXIT MACH NUMBER       = 1.310                      *** /ADDJET
*/   ***         JET EXIT TEMPERATURE       = 560.0                      *** /ADDJET
*/   ***         NOZZLE LIP ANGLE           =  2.05                      *** /ADDJET
*/   ***         JET DEFLECTION ANGLE       =  0.00                      *** /ADDJET
*/   ***                                                                 *** /ADDJET
*/   ***                                                                 *** /ADDJET
*/   ***                                                                 *** /ADDJET
*/   *********************************************************************** /ADDJET
*/   *********************************************************************** /ADDJET
*/
*/
BEGIN GLOBAL DATA
 PID = ISOLATED NACELLE
UID = BHATELEY/HOWELL GENERAL DYNAMICS 817 732 4811
CONFIGURATION = FIRST, SECOND  /TWO PLANES OF SYMMETRY.
MACH = 0.600, CALPHA = 0.0, CBETA = 0.0/ RECORD G5
ALPHA = 0.0
SID =ZERO-DEGREE                          /G6
TOLERANCE = 0.1                                        /G7
SURFACE SELECTION = UPPER                               /G8
SELECTION OF VELOCITY COMPUTATION= BOUNDARY-CONDITION  /G9
PRESSURE COEFFICIENT RULES=ISENTROPIC,SECOND-ORDER        /G12
*/
CHECKOUT  = DIP,1,2,3,   DWG,1,2,4,5             /G17
BEGIN NETWORK DATA
*/  THESE ARE THE NETWORKS FOR THE ISOLATED NACELLE.
 NETWORK=NOSE-CONE-1        4        3        NEW
     0.0000            0.0000            0.0000
```

0.0000	0.0000	0.0000
0.0000	0.0000	0.0000
0.0000	0.0000	0.0000
15.2811	0.0000	3.8100
15.2811	1.9050	3.2996
15.2811	3.2996	1.9050
15.2811	3.8100	0.0000
30.5622	0.0000	7.6200
30.5622	3.8100	6.5991
30.5622	6.5991	3.8100
30.5622	7.6200	0.0000

TRIANGULAR PANEL TOLERANCE = .003
BOUNDARY CONDITION = 1,UPPER
NETWORK=NAC-MID-2 4 10 NEW

30.5622	0.0000	7.6200
30.5622	3.8100	6.5991
30.5622	6.5991	3.8100
30.5622	7.6200	0.0000
40.0000	0.0000	7.6200
40.0000	3.8100	6.5991
40.0000	6.5991	3.8100
40.0000	7.6200	0.0000
49.0000	0.0000	7.6200
49.0000	3.8100	6.5991
49.0000	6.5991	3.8100
49.0000	7.6200	0.0000
58.0000	0.0000	7.6200
58.0000	3.8100	6.5991
58.0000	6.5991	3.8100
58.0000	7.6200	0.0000
67.0000	0.0000	7.6200
67.0000	3.8100	6.5991
67.0000	6.5991	3.8100
67.0000	7.6200	0.0000
76.0000	0.0000	7.6200
76.0000	3.8100	6.5991
76.0000	6.5991	3.8100
76.0000	7.6200	0.0000
85.0000	0.0000	7.6200
85.0000	3.8100	6.5991
85.0000	6.5991	3.8100
85.0000	7.6200	0.0000
94.0000	0.0000	7.6200
94.0000	3.8100	6.5991
94.0000	6.5991	3.8100
94.0000	7.6200	0.0000
103.0000	0.0000	7.6200
103.0000	3.8100	6.5991
103.0000	6.5991	3.8100
103.0000	7.6200	0.0000
112.7760	0.0000	7.6200
112.7760	3.8100	6.5991
112.7760	6.5991	3.8100
112.7760	7.6200	0.0000

TRIANGULAR PANEL TOLERANCE = .003
BOUNDARY CONDITION = 1,UPPER
NETWORK=NOZZLE-3 7 14 NEW

112.7760	0.0000	7.6200
112.7760	1.9722	7.3604
112.7760	3.8100	6.5991
112.7760	5.3882	5.3882
112.7760	6.5991	3.8100
112.7760	7.3604	1.9722
112.7760	7.6200	0.0000
120.0000	0.0000	7.6200
120.0000	1.9722	7.3604
120.0000	3.8100	6.5991
120.0000	5.3882	5.3882
120.0000	6.5991	3.8100
120.0000	7.3604	1.9722
120.0000	7.6200	0.0000
128.0000	0.0000	7.6200
128.0000	1.9722	7.3604
128.0000	3.8100	6.5991
128.0000	5.3882	5.3882
128.0000	6.5991	3.8100
128.0000	7.3604	1.9722
128.0000	7.6200	0.0000
133.0000	0.0000	7.6200
133.0000	1.9722	7.3604
133.0000	3.8100	6.5991
133.0000	5.3882	5.3882
133.0000	6.5991	3.8100
133.0000	7.3604	1.9722
133.0000	7.6200	0.0000
136.0000	0.0000	7.6200
136.0000	1.9722	7.3604
136.0000	3.8100	6.5991
136.0000	5.3882	5.3882
136.0000	6.5991	3.8100
136.0000	7.3604	1.9722
136.0000	7.6200	0.0000
137.1600	0.0000	7.6200
137.1600	1.9722	7.3604
137.1600	3.8100	6.5991
137.1600	5.3882	5.3882
137.1600	6.5991	3.8100
137.1600	7.3604	1.9722
137.1600	7.6200	0.0000
138.3600	0.0000	7.5700
138.3600	1.9593	7.3121
138.3600	3.7850	6.5558
138.3600	5.3528	5.3528
138.3600	6.5558	3.7850
138.3600	7.3121	1.9593
138.3600	7.5700	0.0000
139.4600	0.0000	7.4500
139.4600	1.9282	7.1961
139.4600	3.7250	6.4519
139.4600	5.2679	5.2679
139.4600	6.4519	3.7250
139.4600	7.1961	1.9282
139.4600	7.4500	0.0000
140.5900	0.0000	7.2295

```
     140.5900              1.8711              6.9832
     140.5900              3.6147              6.2609
     140.5900              5.1120              5.1120
     140.5900              6.2609              3.6148
     140.5900              6.9832              1.8711
     140.5900              7.2295              0.0000
     142.0000              0.0000              6.9200
     142.0000              1.7910              6.6842
     142.0000              3.4600              5.9929
     142.0000              4.8932              4.8932
     142.0000              5.9929              3.4600
     142.0000              6.6842              1.7910
     142.0000              6.9200              0.0000
     144.0100              0.0000              6.4839
     144.0100              1.6782              6.2630
     144.0100              3.2419              5.6152
     144.0100              4.5848              4.5848
     144.0100              5.6152              3.2420
     144.0100              6.2630              1.6782
     144.0100              6.4839              0.0000
     147.4400              0.0000              5.7360
     147.4400              1.4846              5.5406
     147.4400              2.8680              4.9675
     147.4400              4.0560              4.0560
     147.4400              4.9675              2.8680
     147.4400              5.5406              1.4846
     147.4400              5.7360              0.0000
     150.8600              0.0000              4.9903
     150.8600              1.2916              4.8203
     150.8600              2.4951              4.3217
     150.8600              3.5287              3.5287
     150.8600              4.3217              2.4952
     150.8600              4.8203              1.2916
     150.8600              4.9903              0.0000
     154.2900              0.0000              4.2424
     154.2900              1.0980              4.0978
     154.2900              2.1212              3.6740
     154.2900              2.9998              2.9998
     154.2900              3.6740              2.1212
     154.2900              4.0978              1.0980
     154.2900              4.2424              0.0000
TRIANGULAR PANEL TOLERANCE = .003
BOUNDARY CONDITION = 1, UPPER
*/
*/                                                                          /ADDJET
*/                                                                          /ADDJET
*/      ***************************************************************     /ADDJET
*/      ***************************************************************     /ADDJET
*/      **                                                           **     /ADDJET
*/      **     THE FOLLOWING NETWORK HAS BEEN NULLIFIED              **     /ADDJET
*/      **                BY THE ADDJET CODE                         **     /ADDJET
*/      **                                                           **     /ADDJET
*/      ***************************************************************     /ADDJET
*/      ***************************************************************     /ADDJET
*/                                                                          /ADDJET
*/                                                                          /ADDJET
*/ NETWORK-EXIT-4         7         2       NEW                             /ADDJET
```

```
*/     154.2900           0.0000          4.2424                                /ADDJET
*/     154.2900           1.0980          4.0978                                /ADDJET
*/     154.2900           2.1212          3.6740                                /ADDJET
*/     154.2900           2.9998          2.9998                                /ADDJET
*/     154.2900           3.6740          2.1212                                /ADDJET
*/     154.2900           4.0978          1.0980                                /ADDJET
*/     154.2900           4.2424          0.0000                                /ADDJET
*/     154.2900           0.0000          0.0                                   /ADDJET
*/     154.2900           0.0000          0.0000                                /ADDJET
*/     154.2900           0.0000          0.0000                                /ADDJET
*/     154.2900           0.0000          0.0000                                /ADDJET
*/     154.2900           0.0000          0.0000                                /ADDJET
*/     154.2900           0.0000          0.0000                                /ADDJET
*/     154.2900           0.0000          0.0000                                /ADDJET
*/TRIANGULAR PANEL TOLERANCE = .003                                             /ADDJET
*/BOUNDARY CONDITION = 1, UPPER                                                 /ADDJET
*/*/                                                                            /ADDJET
*/*/                                                                            /ADDJET
*/                                                                              /ADDJET
*/                                                                              /ADDJET
*/         ************************************************************        /ADDJET
*/         ************************************************************        /ADDJET
*/         **                                                        **        /ADDJET
*/         **   THE FOLLOWING NETWORKS SIMULATE THE EXHAUST            **        /ADDJET
*/         **        PLUME AND THE OUTFLOW VELOCITIES                  **        /ADDJET
*/         **                                                        **        /ADDJET
*/         ************************************************************        /ADDJET
*/         ************************************************************        /ADDJET
*/                                                                              /ADDJET
*/                                                                              /ADDJET
NETWORK=PLUME-SEGMENT-1          7         11                                   /ADDJET
   154.290      0.000      4.242                                                /ADDJET
   154.290      1.098      4.098                                                /ADDJET
   154.290      2.121      3.674                                                /ADDJET
   154.290      3.000      3.000                                                /ADDJET
   154.290      3.674      2.121                                                /ADDJET
   154.290      4.098      1.098                                                /ADDJET
   154.290      4.242      0.000                                                /ADDJET
   156.399      0.000      5.088                                                /ADDJET
   156.399      1.317      4.915                                                /ADDJET
   156.399      2.544      4.406                                                /ADDJET
   156.399      3.598      3.598                                                /ADDJET
   156.399      4.406      2.544                                                /ADDJET
   156.399      4.915      1.317                                                /ADDJET
   156.399      5.088      0.000                                                /ADDJET
   158.508      0.000      5.681                                                /ADDJET
   158.508      1.470      5.488                                                /ADDJET
   158.508      2.841      4.920                                                /ADDJET
   158.508      4.017      4.017                                                /ADDJET
   158.508      4.920      2.841                                                /ADDJET
   158.508      5.488      1.470                                                /ADDJET
   158.508      5.681      0.000                                                /ADDJET
   160.617      0.000      6.127                                                /ADDJET
   160.617      1.586      5.918                                                /ADDJET
   160.617      3.063      5.306                                                /ADDJET
   160.617      4.332      4.332                                                /ADDJET
   160.617      5.306      3.063                                                /ADDJET
```

```
     160.617      5.918      1.586                                          /ADDJET
     160.617      6.127      0.000                                          /ADDJET
     162.726      0.000      6.442                                          /ADDJET
     162.726      1.667      6.222                                          /ADDJET
     162.726      3.221      5.578                                          /ADDJET
     162.726      4.555      4.555                                          /ADDJET
     162.726      5.578      3.221                                          /ADDJET
     162.726      6.222      1.667                                          /ADDJET
     162.726      6.442      0.000                                          /ADDJET
     166.944      0.000      6.687                                          /ADDJET
     166.944      1.731      6.459                                          /ADDJET
     166.944      3.344      5.791                                          /ADDJET
     166.944      4.729      4.729                                          /ADDJET
     166.944      5.791      3.344                                          /ADDJET
     166.944      6.459      1.731                                          /ADDJET
     166.944      6.687      0.000                                          /ADDJET
     171.163      0.000      6.466                                          /ADDJET
     171.163      1.674      6.246                                          /ADDJET
     171.163      3.233      5.600                                          /ADDJET
     171.163      4.572      4.572                                          /ADDJET
     171.163      5.600      3.233                                          /ADDJET
     171.163      6.246      1.674                                          /ADDJET
     171.163      6.466      0.000                                          /ADDJET
     179.599      0.000      6.001                                          /ADDJET
     179.599      1.553      5.796                                          /ADDJET
     179.599      3.000      5.197                                          /ADDJET
     179.599      4.243      4.243                                          /ADDJET
     179.599      5.197      3.000                                          /ADDJET
     179.599      5.796      1.553                                          /ADDJET
     179.599      6.001      0.000                                          /ADDJET
     188.035      0.000      5.807                                          /ADDJET
     188.035      1.503      5.609                                          /ADDJET
     188.035      2.903      5.029                                          /ADDJET
     188.035      4.106      4.106                                          /ADDJET
     188.035      5.029      2.903                                          /ADDJET
     188.035      5.609      1.503                                          /ADDJET
     188.035      5.807      0.000                                          /ADDJET
     196.472      0.000      5.757                                          /ADDJET
     196.472      1.490      5.561                                          /ADDJET
     196.472      2.879      4.986                                          /ADDJET
     196.472      4.071      4.071                                          /ADDJET
     196.472      4.986      2.879                                          /ADDJET
     196.472      5.561      1.490                                          /ADDJET
     196.472      5.757      0.000                                          /ADDJET
     204.908      0.000      5.718                                          /ADDJET
     204.908      1.480      5.523                                          /ADDJET
     204.908      2.859      4.952                                          /ADDJET
     204.908      4.043      4.043                                          /ADDJET
     204.908      4.952      2.859                                          /ADDJET
     204.908      5.523      1.480                                          /ADDJET
     204.908      5.718      0.000                                          /ADDJET
TRIANGULAR PANEL TOLERANCE = .003                                           /ADDJET
BOUNDARY CONDITION = 2,UPPER                                                /ADDJET
SPECIFIED FLOW                                                              /ADDJET
TERM = 1                                                                    /ADDJET
POINTS = ALL  $ 0.0                                                         /ADDJET
*/                                                                          /ADDJET
```


C-3

ORIGINAL PAGE IS
OF POOR QUALITY

```
*/                                                                       /ADDJET
*/       THE FOLLOWING DATA ARE THE OUTFLOW VELOCITIES                   /ADDJET
*/       THROUGH THE PANELS OF THE PLUME NETWORK                         /ADDJET
POINTS = CENTER                                                          /ADDJET
    -0.037     -0.037     -0.037     -0.037     -0.037     -0.037        /ADDJET
    -0.026     -0.026     -0.026     -0.026     -0.026     -0.026        /ADDJET
    -0.022     -0.022     -0.022     -0.022     -0.022     -0.022        /ADDJET
    -0.029     -0.029     -0.029     -0.029     -0.029     -0.029        /ADDJET
    -0.038     -0.038     -0.038     -0.038     -0.038     -0.038        /ADDJET
    -0.042     -0.042     -0.042     -0.042     -0.042     -0.042        /ADDJET
    -0.048     -0.048     -0.048     -0.048     -0.048     -0.048        /ADDJET
    -0.043     -0.043     -0.043     -0.043     -0.043     -0.043        /ADDJET
    -0.036     -0.036     -0.036     -0.036     -0.036     -0.036        /ADDJET
    -0.032     -0.032     -0.032     -0.032     -0.032     -0.032        /ADDJET
*/                                                                       /ADDJET
*/                                                                       /ADDJET
*/                                                                       /ADDJET
*/                                                                       /ADDJET
*/         *****************************************************         /ADDJET
*/         *****************************************************         /ADDJET
*/         **                                                 **         /ADDJET
*/         **       THE FOLLOWING NETWORKS FORM THE BASE      **         /ADDJET
*/         **              OF THE EXHAUST PLUME               **         /ADDJET
*/         **                                                 **         /ADDJET
*/         *****************************************************         /ADDJET
*/         *****************************************************         /ADDJET
*/                                                                       /ADDJET
*/                                                                       /ADDJET
NETWORK=PLUME-BASE-SEG-1          7        2                             /ADDJET
   204.908     0.000     5.718                                           /ADDJET
   204.908     1.480     5.523                                           /ADDJET
   204.908     2.859     4.952                                           /ADDJET
   204.908     4.043     4.043                                           /ADDJET
   204.908     4.952     2.859                                           /ADDJET
   204.908     5.523     1.480                                           /ADDJET
   204.908     5.718     0.000                                           /ADDJET
   204.908     0.000     0.000                                           /ADDJET
   204.908     0.000     0.000                                           /ADDJET
   204.908     0.000     0.000                                           /ADDJET
   204.908     0.000     0.000                                           /ADDJET
   204.908     0.000     0.000                                           /ADDJET
   204.908     0.000     0.000                                           /ADDJET
   204.908     0.000     0.000                                           /ADDJET
TRIANGULAR PANEL TOLERANCE = .003                                        /ADDJET
BOUNDARY CONDITION =  4,  8,1    6,3                                     /ADDJET
SINGULARITY TYPES =  SA, DA                                              /ADDJET
*/                                                                       /ADDJET
*/                                                                       /ADDJET
*/         *****************************************************         /ADDJET
*/         *****************************************************         /ADDJET
*/         **                                                 **         /ADDJET
*/         **       THE FOLLOWING NETWORKS FORM THE WAKE      **         /ADDJET
*/         **            FROM THE EXHAUST PLUME               **         /ADDJET
*/         **                                                 **         /ADDJET
*/         *****************************************************         /ADDJET
*/         *****************************************************         /ADDJET
*/                                                                       /ADDJET
```

```
*/                                                                          /ADDJET
NETWORK=PLUME-WAKE-SEG-1          2          7                              /ADDJET
   204.908      0.000      5.718                                            /ADDJET
   457.997      0.000      5.718                                            /ADDJET
   204.908      1.480      5.523                                            /ADDJET
   457.997      1.480      5.523                                            /ADDJET
   204.908      2.859      4.952                                            /ADDJET
   457.997      2.859      4.952                                            /ADDJET
   204.908      4.043      4.043                                            /ADDJET
   457.997      4.043      4.043                                            /ADDJET
   204.908      4.952      2.859                                            /ADDJET
   457.997      4.952      2.859                                            /ADDJET
   204.908      5.523      1.480                                            /ADDJET
   457.997      5.523      1.480                                            /ADDJET
   204.908      5.718      0.000                                            /ADDJET
   457.997      5.718      0.000                                            /ADDJET
WAKE FLOW PROPERTIES TAG                                                    /ADDJET
BOUNDARY CONDITION =  1,  WAKE 1                                            /ADDJET
BEGIN GEOMETRIC EDGE MATCHING
ABUTMENT = 2,2  ENTIRE EDGE  +
         = 3,4  ENTIRE EDGE
*/
*/
*/
BEGIN FLOW PROPERTIES DATA
SURFACE FLOW PROPERTIES = CASE-1
NETWORK-IMAGES =1=2=3=4=5=6
POINTS = CENTER
PRINTOUT = 1,2,8,13,16
*/
*/
FORCES AND MOMENTS
REFERENCE PARAMETERS = SR, 182.0, BR, 1.0, CR, 1.0
AXIS SYSTEM = RCS,0.,0.,0., WAS, 0.,0.,0.
PRINTOUT =   NETWORK, CONFIGURATION
CASE = CASE-1
NETWORK-IMAGES =1=2=3
SURFACE = UPPER
END PROBLEM DEFINITION
**EOR**
**EOF**
FOR.
```

```
      PROGRAM ADDJET                              ORIGINAL PAGE IS
C                                                 OF POOR QUALITY
C
C
C     PROGRAM ADDJET DEVELOPS THE OUTPUT FROM WILMOTH'S RAXJET CODE
C     INTO NETWORKS THAT CAN BE USED BY PANAIR.  ADDJET IS THE MASTER
C     CONTROL SUBROUTINE FOR THIS PROGRAM.
C
C
      COMMON /TITL/    TITLE
      COMMON /DUMP/ IDMP(30)
      COMMON /PADAT/  OLDFIL,NEWFIL, ENAME,ABUT(6),IEDGE(6)
      COMMON /RAXIN/ XNPR,EMJET,TTJET,THLIP,THETAJ,JDUM,IPRINT
     1              ,AMINF
      COMMON /NETWORK/ NETWORK,M,N,X(20,20),Y(20,20),Z(20,20),
     1                 BC,IFLAG
      COMMON /CLINE/ XC,YC,ZC,A,B,C,EAREA,DIA
      COMMON /PLUME/ MP(6),NP(6),XP(6,20,20),YP(6,20,20),
     1               ZP(6,20,20),VP(6,19,19)
      COMMON /RAXOUT/ NPTS,XSTA(30),XRATIO(30),XVN(30)
      COMMON /PBASE/ MPB(6),NPB(6),XPB(6,20,2),YPB(6,20,2),
     1               ZPB(6,20,2)
      COMMON /CUT/ NCUT,CUT(11)
      COMMON /WAKE/ MW(6),NW(6),XW(6,2,20),YW(6,2,20),
     1              ZW(6,2,20)
C
C
      CHARACTER  TITLE*80,OLDFIL*20,NEWFIL*20,LINE*80, ENAME*20,
     1           ABUT*20,NETWORK*20,BC*4
C
C
      NAMELIST /PADATN/ OLDFIL,NEWFIL, ENAME,ABUT,IEDGE
      NAMELIST /RAXINN/ XNPR,EMJET,TTJET,THLIP,THETAJ,JDUM,IPRINT
     1                 ,AMINF
C
C
C
      CALL ASSIGN(11,'11')
C*********  READ INPUT DATA  ***********************************
C
      OPEN(UNIT=3,FILE='ADDJET.INP',
     1     STATUS='OLD')
      READ(3,80,END=120) TITLE
 80   FORMAT(1X,A80)
C
      READ(3,85) (IDMP(I), I= 1,20)
 85   FORMAT(20I1)
C
      GO TO 140
 120  WRITE(11,130)
 130  FORMAT(1H1,10X,'END OF FILE ENCOUNTERED ON INPUT FILE')
C
C
C
 140  WRITE(11,90) TITLE
 90   FORMAT(10X,A80)
```

```
      READ(3,PADATN)
      READ(3,RAXINN)
C
      CLOSE(UNIT=3,DISPOSE='KEEP')
C
      OPEN(UNIT=9,FILE=OLDFIL,STATUS='OLD')
      OPEN(UNIT=10,FILE=NEWFIL,STATUS='NEW',
     1          CARRIAGE CONTROL='LIST')
C
C
C
C************  READ PAN AIR DATA FILE  ****************************
      IFLAG = 0
 155  CALL READNET
      IF(IFLAG.EQ.2) GO TO 170
C
C
C
C***********  DEFINE THE PLUME CENTERLINE  *************************
C
C
      IF(NETWORK.EQ.ENAME) THEN
         CALL CENTER
         CALL CLINE
      END IF
C
C
C*********   DETERMINE THE COORDINATES FOR THE UPSTREAM     ***********
C*********            EDGE OF THE PLUME                    ************
C
C
C
      DO NET = 1,6
         IF(NETWORK.EQ.ABUT(NET)) THEN
            CALL START(NET)
            NPLUME = NET
         END IF
      END DO
      IF(IFLAG.LT.2) GO TO 155
C
 170  CONTINUE
      WRITE(11,710)
 710  FORMAT(/,20X,'ALL DATA HAVE BEEN READ FROM THE PANAIR DECK',//)
C
C
C***********      EXTEND THE PLUME DOWNSTREAM    *********************
C
C          CALL THE RAXJET CODE TO OBTAIN THE JET EXPANSION RATIO IN
C     THE DOWNSTREAM DIRECTION AND THE CORRESPONDING INFLOW VELOCITIES
C     VELOCITIES.  THE VARIABLES ARE PUT IN THE COMMON BLOCK /RAXOUT/.
C
      IF(JDUM.EQ.1) CALL RAXSUB
      IF(JDUM.EQ.0) CALL RAXMAIN
C
C     CALL SUBROUTINE "EXTEND" TO EXTEND THE PLUMES DOWNSTREAM
C     FROM THE EXIT PLANE.
C
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
         CALL EXTEND
C
C
C
C
C******************** DEFINE THE PLUME BASE NETWORKS ***********
C
      CALL PBASE
      CALL WAKE
C
C
C**************** CREATE THE NEW PAN AIR FILE ***************
C
      CALL CREATE
C
      END
```

```
      SUBROUTINE READNET
C
C
C     SUBROUTINE 'READNET' READS THE PANAIR FILE TO
C     DEFINE THE NEXT NETWORK.
C
C
      COMMON /DUMP/ IDMP(30)
      COMMON /PADAT/ OLDFIL,NEWFIL,ENAME,ABUT(6),IEDGE(6)
      COMMON /NETWORK/  NETWORK,M,N,X(20,20),Y(20,20),Z(20,20),
     1            BC,IFLAG
      CHARACTER LINE*80, BC*4, NETWORK*20, ABUT*20
C
C**********  FIND WHERE THE NETWORK DATA BEGINS  **************
C
C
 200  READ(9,210) LINE
 210  FORMAT(A80)
C
      IF(LINE(1:2).EQ.'*/') GO TO 200
C
      IF(IFLAG.EQ.0) THEN
         DO  I = 1, 20
            J = I + 3
            IF(LINE(I:J).EQ.'BEG1') GO TO 220
         END DO
      GO TO 200
      END IF
C
C
      IF(IFLAG.EQ.1) THEN
         DO I = 1,20
            J= I + 3
             IF(LINE(I:J).EQ.'BEG1')THEN
                 IFLAG =2
                GO TO 390
               END IF
          END DO
       END IF
C
C
C
 220  CONTINUE
      DO I = 1,30
         J = I + 3
         IF(LINE(I:J).EQ.'NETW'.AND.IFLAG.EQ.0) GO TO 230
         IF(LINE(I:J).EQ.'NETW'.AND.IFLAG.EQ.1) GO TO 235
      END DO
      GO TO 200
C
C

C***********  FIND THE NETWORK NAME CARD  ***************

C
 230  CONTINUE
      READ(9,210) LINE
```

C IGINAL PAGE IS
OF POOR QUALITY

```
 235   CONTINUE
       IF(LINE(1:2).EQ.'*/') GO TO 230
       DO I = 1,30
          J = I + 3
          IF(LINE(I:J).EQ.'NETW') GO TO 236
          IF(LINE(I:J).EQ.'GEOM')  THEN
             WRITE(11,*) 'FOUND THE NEXT BEGIN  CARD'
             FLAG = 2.
             GO TO 380
          END IF
       END DO
       GO TO 230
C
C
C
C*********** READ THE NETWORK DATA ***********************
 236   CONTINUE
C
C      NOW FIND THE CHARACTERS  "NETWORK ="
       IF(LINE(1:2).EQ.'*/') GO TO 236
       DO I = 1,20
          IF(LINE(I:I).EQ.'=') GO TO 238
       END DO
C
C      NOW FIND START OF THE NETWORK NAME
 238   N1 = I + 1
       DO J = I+1 , 50
          IF(LINE(J:J) NE.' ') GO TO 240
          N1 = N1 + 1
       END DO
C
C
C      FIND END OF THE NETWORK NAME
 240   N2 = N1
       DO I = J, 50
          IF(LINE(I:I).EQ.' ') THEN
             N2 = I - 1
             GO TO 250
          END IF
       END DO
 250   NETWORK = LINE(N1:N2)
       GO TO 260
C
C
C
C
C************  READ THE VALUE OF M    *************************
C
 260   M = 0
       DO I=  N2+1 ,80
          IF(LINE(I:I).NE.' ') THEN
             READ(LINE(I:I),270,ERR=360) M
 270         FORMAT(I1)
             N3 = I + 1
             IF(LINE(N3:N3).NE.' ') THEN
                READ(LINE(I:I+1),280,ERR=360) M
 280            FORMAT(I2)
```

```
               N3 = I + 2
            END IF
         END IF
         IF(M.GT.0) GO TO 285
         IF(I.GE.79) GO TO 200
      END DO
C
C
C*************** READ VALUE OF N  *********************************
C
 285  N  =  0
      DO I = N3, 80
         IF(LINE(I:I).NE.' ') THEN
            READ(LINE(I:I),290,ERR=360) N
 290        FORMAT(I1)
            IF(LINE(I+1:I+1).NE.' ') THEN
               READ(LINE(I:I+1),300,ERR=360) N
 300           FORMAT(I2)
            END IF
         END IF
         IF(I.GE.79) GO TO 200
         IF(N.GT.0) GO TO 310
      END DO
C
C
C
C************  READ THE X,Y,Z  COORDINATES  ***********************
C
 310  READ(9,*) ((X(I,J),Y(I,J),Z(I,J), I=1,M), J=1,N)
C
C
C**************  READ THE BOUNDARY CONDITION  ************************
C
 500  CONTINUE
      READ(9,510) LINE
 510  FORMAT(A80)
C
      DO I = 1,10
         J = I + 3
         IF(LINE(I:J).EQ.'BOUN') GO TO 320
         IF(LINE(I:J).EQ.'NETW') THEN
         WRITE(11,530) NETWORK
 530  FORMAT(///,30X,'THE BOUNDARY CONDITION WAS NOT FOUND ',
     1               'FOR NETWORK',
     2               A20,/,30X,'PROGRAM WILL BE TERMINATED.')
C
            STOP
         END IF
      END DO
      GO TO 500
C
C
 320  CONTINUE
      DO I = 1,76
         J = I + 3
         IF(LINE(I:J).EQ.'UPPE') GO TO 325
         IF(LINE(I:J).EQ.'LOWE') GO TO 325
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
            IF(LINE(I:J).EQ.'AVER') GO TO 325
            IF(LINE(I:J).EQ.'WAKE') GO TO 325
         END DO
C
         DO I = 1,6
            IF(NETWORK.EQ.ABUT(I))  WRITE(11,520)  NETWORK
         END DO
         GO TO 325
 520     FORMAT(//,20X,'THE BOUNDARY CONDITION ON NETWORKS ABUTTING THE',/,
     1      20X,'PLUME MUST BE CLASS 1, SUBCLASS UPPER OR LOWER',/,20X,
     2      'IF THE SUBCLASS INDEX WAS USED INSTEAD OF THE KEYWORD',/,20X,
     3      'THEN CHANGE TO THE KEYWORD DESIGNATION.  THIS MUST BE DONE',
     4.     /,20X,'FOR NETWORK ',A20,/,20X,'SINCE IT ABUTS THE PLUME.',//)
C
 325     CONTINUE
         BC = LINE(I:J)
         GO TO 370
C
C
 360     CONTINUE
         WRITE(11,362) LINE
 362     FORMAT(20X,'END OF CARD REACHED WHILE ATTEMPTING',
     1      ' TO READ THE VALUES OF M AND N ON THE FOLLOWING CARD',
     2          /,A80)
         STOP
C
C
C
C
C
 370     CONTINUE
         IFLAG = 1
C
 380     CONTINUE
         IF(IDMP(2).EQ.1) WRITE(11,910) NETWORK,M,N,BC,IFLAG
 910     FORMAT(/5X,'DUMP FROM SUBROUTINE READNET',
     1      ' ***********************************************',/,
     1          10X,'NETWORK = ',A20,/
     2          10X,'       M = ',I4,/
     3          10X,'       N = ',I4,/
     4          10X,'      BC =    ',A4,/
     5          10X,'   IFLAG =  ',I4)
 390     CONTINUE
         RETURN
         END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE CENTER
C
C      PROGRAM CENTER LOCATES THE GEOMETRIC CENTER OF THE
C      EXIT NETWORK AND COMPUTES THE EXIT AREA
C
C      THE OPTIONS AVAILABLE TO DO THIS ARE:
C
C           ICENTR = 1   THE CENTER OF THE EXIT IS COMPUTED AS THE
C                        AVERAGE OF THE FOUR CORNER POINTS.
C
C           ICENTR = 2   THE CENTER OF THE EXIT WILL BE PROVIDED
C                        BY THE USER.
C
C     INPUT COMMON BLOCKS  =  /NETWORK/  AND  /PADAT/
      COMMON /DUMP/ IDMP(30)
      COMMON /PADAT/OLDFIL,NEWFIL,ENAME,ABUT(6),IEDGE(6)
      COMMON /NETWORK/ NETWORK,M,N,X(20,20),Y(20,20),Z(20,20),
     1                      BC,IFLAG
C
C     OUTPUT COMMON BLOCKS = /CLINE/
      COMMON /CLINE/ XC,YC,ZC,A,B,C,EAREA,DIA
C
      CHARACTER  NETWORK*20,BC*4,ENAME*20,OLDFIL*20,NEWFIL*20
C
      IF(IDMP(3).EQ.1) WRITE(11,705)
 705  FORMAT(//,5X,'DUMP FORM SUBROUTINE CENTER ***********',
     1      '**************************')
C
C*************  CHECK SYMMETRY IN Y=0 PLANE  **************************
C
C     (I.E. IF ALL VALUES OF Y ARE ZERO THEN THAT EDGE
C      ABUTTS THE PLANE OF SYMMETRY.)
C
      Y1 = 0.
      Y2 = 0.
      Y3 = 0.
      Y4 = 0.
C
      DO I = 1,M
         Y4 = Y4 + Y(I,1)
         Y2 = Y2 + Y(I,N)
      END DO
C
      DO J = 1,N
         Y1 = Y1 + Y(1,J)
         Y3 = Y3 + Y(M,J)
      END DO
C
      IF(Y1.EQ.0.0.AND.IDMP(3).EQ.1) WRITE(11,710)
      IF(Y2.EQ.0.0.AND.IDMP(3).EQ.1) WRITE(11,720)
      IF(Y3.EQ.0.0.AND.IDMP(3).EQ.1) WRITE(11,730)
      IF(Y4.EQ.0.0.AND.IDMP(3).EQ.1) WRITE(11,740)
C
      IF(Y1.EQ.0.0) ISYM = 1
      IF(Y2.EQ.0.0) ISYM = 1
      IF(Y3.EQ.0.0) ISYM = 1
      IF(Y4.EQ.0.0) ISYM = 1
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C
 710   FORMAT(20X, 'SIDE 1 OF THE EXIT PLANE ABUTS THE ',
     1                    'PLANE OF SYMMETRY, Y=0')
 720   FORMAT(20X, 'SIDE 2 OF THE EXIT PLANE ABUTS THE ',
     1                    'PLANE OF SYMMETRY, Y=0')
 730   FORMAT(20X, 'SIDE 3 OF THE EXIT PLANE ABUTS THE ',
     1                    'PLANE OF SYMMETRY, Y=0')
 740   FORMAT(20X, 'SIDE 4 OF THE EXIT PLANE ABUTS THE ',
     1                    'PLANE OF SYMMETRY, Y=0')
C
C
       XC = (X(1,1)+X(M,1)+X(1,N)+X(M,N)) / 4.0
       YC = (Y(1,1)+Y(M,1)+Y(1,N)+Y(M,N)) / 4.0
       ZC = (Z(1,1)+Z(M,1)+Z(1,N)+Z(M,N)) / 4.0
C
       IF(ISYM.EQ.1) YC = 0.0
C
C*************  CHECK SYMMETRY IN Z=0 PLANE  *************************
C
C      (I.E. IF ALL VALUES OF Z ARE ZERO THEN THAT EDGE
C       ABUTTS THE PLANE OF SYMMETRY.)
       JSYM = 0
C
       Z1 = 0.
       Z2 = 0.
       Z3 = 0.
       Z4 = 0.
C
       DO I = 1,M
          Z4 = Z4 + Z(I,1)
          Z2 = Z2 + Z(I,N)
       END DO
C
       DO J = 1,N
          Z1 = Z1 + Z(1,J)
          Z3 = Z3 + Z(M,J)
       END DO
C
       IF(Z1.EQ.0.0.AND.IDMP(3).EQ.1) WRITE(11,810)
       IF(Z2.EQ.0.0.AND.IDMP(3).EQ.1) WRITE(11,820)
       IF(Z3.EQ.0.0.AND.IDMP(3).EQ.1) WRITE(11,830)
       IF(Z4.EQ.0.0.AND.IDMP(3).EQ.1) WRITE(11,840)
C
       IF(Z1.EQ.0.0) JSYM = 1
       IF(Z2.EQ.0.0) JSYM = 1
       IF(Z3.EQ.0.0) JSYM = 1
       IF(Z4.EQ.0.0) JSYM = 1
C
 810   FORMAT(20X, 'SIDE 1 OF THE EXIT PLANE ABUTS THE ',
     1                    'PLANE OF SYMMETRY, Z=0')
 820   FORMAT(20X, 'SIDE 2 OF THE EXIT PLANE ABUTS THE ',
     1                    'PLANE OF SYMMETRY, Z=0')
 830   FORMAT(20X, 'SIDE 3 OF THE EXIT PLANE ABUTS THE ',
     1                    'PLANE OF SYMMETRY, Z=0')
 840   FORMAT(20X, 'SIDE 4 OF THE EXIT PLANE ABUTS THE ',
     1                    'PLANE OF SYMMETRY, Z=0')
C
```

```
      IF(JSYM.EQ.1) ZC = 0.0
C
C
C
C       IF ICENTR = 2 THEN XCL(1), YCL(1), AND ZCL(1)
C       MUST BE PROVIDED BY USER------ADD CODE FOR THIS!!
C
C
C
C*******  COMPUTE THE AREA OF THE NOZZLE EXIT WITH THE **************
C*******  MENSURATION FORMULA FOR A GENERAL QUADRILATERAL ************
C         THE AREA OF EACH PANEL WILL BE COMPUTED INDIVIDUALLY AND
C         SUMMED TO GET THE AREA OF THE EXIT.
C
 110  CONTINUE
      EAREA = 0.0
C
C
      IF(IDMP(3).EQ.1) WRITE(11,901)
 901  FORMAT(//,20X,
     1 'I  J  X(I,J) Y(I,J) Z(I,J)   ASQ   BSQ   CSQ'
     2 '     DSQ     PSQ      QSQ     EAREA')
      DO J = 1, N-1
      DO I = 1, M-1
         II = I+1
         JJ = J+1
      ASQ=(X(I,JJ)-X(I,J))**2 +(Y(I,JJ)-Y(I,J))**2 +
     1              (Z(I,JJ)-Z(I,J))**2
      BSQ=(X(II,JJ)-X(I,JJ))**2+(Y(II,JJ)-Y(I,JJ))**2+
     1              (Z(II,JJ)-Z(I,JJ))**2
      CSQ=(X(II,J)-X(II,JJ))**2+(Y(II,J)-Y(II,JJ))**2+
     1              (Z(II,J)-Z(II,JJ))**2
      DSQ=(X(I,J)-X(II,J))**2+(Y(I,J)-Y(II,J))**2+
     1              (Z(I,J)-Z(II,J))**2
      PSQ=(X(II,J)-X(I,JJ))**2+(Y(II,J)-Y(I,JJ))**2+
     1              (Z(II,J)-Z(I,JJ))**2
      QSQ=(X(II,JJ)-X(I,J))**2+(Y(II,JJ)-Y(I,J))**2+
     1              (Z(II,JJ)-Z(I,J))**2
C
      EAREA= EAREA + 0.25*SQRT(4.*PSQ*QSQ-(BSQ+DSQ-ASQ-CSQ)**2)
C
C
C
C
      IF(IDMP(3).EQ.1) WRITE(11,905) I,J,X(I,J),Y(I,J),Z(I,J),ASQ,
     1              BSQ,CSQ,DSQ,PSQ,QSQ,EAREA
 905  FORMAT(18X,2I3,10F7.2)
      END DO
      END DO
C
      IF(IDMP(3).EQ.1)  WRITE(11,750) NETWORK
      IF(ISYM.EQ.0.AND.JSYM.EQ.0.AND.IDMP(3).EQ.1) WRITE(11,760) EAREA
      IF(ISYM.EQ.1.AND.JSYM.EQ.0) THEN
         EAREA = 2. * EAREA
         IF(IDMP(3).EQ.1)  WRITE(11,770) EAREA
      END IF
      IF(ISYM.EQ.0.AND.JSYM.EQ.1) THEN
```

ORIGINAL PAGE IS OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

```
          EAREA = 2. * EAREA
           IF(IDMP(3).EQ.1)  WRITE(11,780) EAREA
        END IF
        IF(ISYM.EQ.1.AND.JSYM.EQ.1) THEN
           EAREA = 4. * EAREA
           IF(IDMP(3).EQ.1)  WRITE(11,790) EAREA
        END IF
C
C
C        COMPUTE THE EQUIVALENT NOZZLE EXIT DIAMETER
        DIA = SQRT(4. * EAREA /3.14159)
C
C
  750   FORMAT(/,20X,'THE NOZZLE EXIT NETWORK IS ',A20)
  760   FORMAT(20X,'THE EXIT AREA IS ',F10.3)
  770   FORMAT(20X,'THIS NETWORK ABUTS THE "Y" PLANE OF SYMMETRY',
      1     /,20X,'THE TOTAL EXIT AREA IS ',F10.3)
  780   FORMAT(20X,' THIS NETWORK ABUTS THE "Z" PLANE OF SYMMETRY',
      1  /,20X,'THE TOTAL EXIT AREA IS ',F10.3)
  790   FORMAT(20X,'THIS NETWORK ABUTS THE "Y" AND "Z" PLANES OF ',
      1    'SYMMETRY',/,20X,'THE TOTAL EXIT AREA IS ',F10.3)
C
C
        IF(IDMP(3).EQ.1) WRITE(11,910) XC,YC,ZC,EAREA,DIA
  910   FORMAT(//,
      1           10X,'     XC = ',F8.3,
      2           10X,'     YC = ',F8.3,/,
      3           10X,'     ZC = ',F8.3,/,
      4           10X,' EAREA = ',F8.3,/,
      5           10X,'    DIA = ',F8.3)
C
        RETURN
        END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE CLINE
C
C
C       SUBROUTINE 'CLINE' DETERMINES THE COEFFICIENTS FOR AN
C       EQUATION THAT DESCRIBES THE CENTERLINE PATH OF THE PLUME.
C       THE INITIAL DEFLECTION ANGLE IS ACCOUNTED FOR.
C
C       THE FOLLOWING ASSUMPTIONS ARE MADE IN DEVELOPING THE
C       PATH OF THE PLUME CENTERLINE.
C
C              1. THE PLUME STARTS AT THE CENTERLINE OF THE EXIT NETWORK.
C
C              2.THE INITIAL SLOPE OF THE PATH IS THE SAME AS THE
C                JET DEFLECTION ANGLE.
C
C              3. THE PLUME EXTENDS DOWNSTREAM FOR A DISTANCE OF 10
C                 NOZZLE EXIT DIAMETERS, AT WHICH POINT THE SLOPE IS ZERO
C
C              4. THE Y VALUE OF THE PATH DOES NOT CHANGE.
C
C              5. THE EQUATION OF THE PATH TAKES THE FOLLOWING
C                 FORM WHERE Z AND X ARE INCREMENTAL VALUES
C                 RELATIVE TO THE PLUME CENTERLINE.
C                  Z = A*X**2  + B*X  + C
C
C
      COMMON /DUMP/ IDMP(30)
      COMMON /RAXIN/ XNPR, EMJET,TTJET,THLIP,THETAJ,JDUM,IPRINT
     1              ,AMINF
      COMMON /CLINE/ XC,YC,ZC,A,B,C,EAREA,DIA
C
      X1 = 10. * DIA
C
      THETA = - THETAJ /57.2958
      A = - TAN(THETA) /(2. * X1)
      B = TAN(THETA)
      C = 0.0
C
C
      IF(IDMP(4).EQ.1) WRITE(11,910) A,B,C
  910 FORMAT(/5X,'DUMP FROM SUBROUTINE CLINE',
     1   ' *****************************************',/,
     1           10X,'COEFFICIENTS OF THE PLUME CENTERLINE ',
     2           'EQUATION ARE:',/,10X,'A = ',F8.4,/,10X,
     3           'B = ',F8.4,/,10X,'C = ',F8.4)
      RETURN
      END
```

```
      SUBROUTINE START(NET)
C
C     SUBROUTINE START READS DATA FOR THE NETWORKS THAT ABUT THE EXIT
C     NETWORK.  THE COORDINATES OF THE ABUTTING EDGES DEFINE THE
C     INITIAL EDGE POINTS OF THE PLUME.
C
C
C
C     THE LEADING EDGE OF THE PLUME WILL ALWAYS BE EDGE 4, AND
C     THE BOUNDARY CONDITION WILL ALWAYS BE CLASS 2, UPPER.
C
C
C
      COMMON /DUMP/ IDMP(30)
        COMMON /PADAT/ OLDFIL,NEWFIL,ENAME,ABUT(6),IEDGE(6),IBUG
      COMMON/NETWORK/ NETWORK,M,N,X(20,20),Y(20,20),Z(20,20),
     1                 BC,IFLAG
      COMMON /CLINE/ XC,YC,ZC,A,B,C,EAREA
      COMMON /PLUME/ MP(6),NP(6),XP(6,20,20),YP(6,20,20),
     1                ZP(6,20,20),VP(6,20)
      CHARACTER NETWORK*20,BC*4,OLDFIL*20,NEWFIL*20,ENAME*20,ABUT*20
C
C
C
      IF(IEDGE(NET).EQ.1.AND.BC.EQ.'UPPE') THEN
         MP(NET) = N
         DO J = 1,N
            XP(NET,J,1) = X(1,J)
            YP(NET,J,1) = Y(1,J)
            ZP(NET,J,1) = Z(1,J)
         END DO
      END IF
C
      IF(IEDGE(NET).EQ.1.AND.BC.EQ.'LOWE') THEN
         MP(NET) = N
         DO J = 1,N
            K = N - J + 1
            XP(NET,J,1) = X(1,K)
            YP(NET,J,1) = Y(1,K)
            ZP(NET,J,1) = Z(1,K)
         END DO
      END IF
C
      IF(IEDGE(NET).EQ.2.AND.BC.EQ.'UPPE') THEN
         MP(NET) = M
         DO I = 1,M
            XP(NET,I,1) = X(1,N)
            YP(NET,I,1) = Y(1,N)
            ZP(NET,I,1) = Z(I,N)
            END DO
      END IF
C
      IF(IEDGE(NET).EQ.2.AND.BC.EQ.'LOWE') THEN
         MP(NET) = M
         DO I = 1,M
            K = M - I + 1
            XP(NET,I,1) = X(K,N)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
            YP(NET,1,1) = Y(K,N)
            ZP(NET,I,1) = Z(K,N)
         END DO
      END IF
C
      IF(IEDGE(NET).EQ.3.AND.BC.EQ.'UPPE') THEN
         MP(NET) = N
         DO J = 1,N
            K = N - J + 1
            XP(NET,J,1) = X(M,K)
            YP(NET,J,1) = Y(M,K)
            ZP(NET,J,1) = Z(M,K)
         END DO
      END IF
C
      IF(IEDGE(NET).EQ.3.AND.BC.EQ.'LOWE') THEN
         MP(NET) = N
         DO J = 1,N
            XP(NET,J,1) = X(M,J)
            YP(NET,J,1) = Y(M,J)
            ZP(NET,J,1) = Z(M,J)
         END DO
      END IF
C
      IF(IEDGE(NET).EQ.4.AND.BC.EQ.'UPPE') THEN
         MP(NET) = M
         DO I = 1,M
            K = M - I + 1
            XP(NET,I,1) = X(K,1)
            YP(NET,I,1) = Y(K,1)
            ZP(NET,I,1) = Z(K,1)
         END DO
      END IF
C
      IF(IEDGE(NET).EQ.4.AND.BC.EQ.'LOWE') THEN
         MP(NET) = M
         DO I = 1,M
            XP(NET,I,1) = X(1,1)
            YP(NET,I,1) = Y(I,1)
            ZP(NET,I,1) = Z(1,1)
         END DO
      END IF
C
C
C
      IF(IDMP(5).EQ.1) WRITE(11,910) NETWORK,NET,(XP(NET,1,1),
     1      YP(NET,1,1),ZP(NET,1,1),   1 = 1,MP(NET))
 910  FORMAT(/5X,'DUMP FROM SUBROUTINE START',
     1   ' *******************************************',/,
     1   10X,'THE EDGE OF THE PLUME SEGMENT FOLLOWS THAT ABUTS TO THE ',
     2   'NETWORK =',A20,/,10X,'IT IS PLUME PART NUMBER ',I2,/,
     3   10X,'      XP       YP          ZP',/,(10X,3F8.3))
      RETURN
      END
```

```
      SUBROUTINE EXTEND
C
C          SUBROUTINE 'EXTEND' IS SUPPLIED WITH M COORDINATES ALONG
C     THE FORWARD EDGE OF THE PLUME IN THE VARIABLE XP(NET,M,1) ETC..
C     ...... WHERE NET INDICATES THE (NET)TH NETWORK OF THE PLUME.  IT
C     IS ALSO SUPPLIED WITH THE COEFFICIENTS OF AN EQUATION
C     DEFINING THE INCREMENTAL VARIATION IN PLUME HEIGHT DOWNSTREAM
C     OF THE EXIT (A,B).
C
C          THE PLUME WILL BE PANELLED SUCH THAT THE M DIRECTION (EDGE 1)
C     ABUTTS THE NACELLE AND THE N DIRECTION IS DOWNSTREAM.  THE
C     'UPPER' SURFACE WILL BE EXPOSED TO THE FLOW.
C
C         THE PLUME WILL EXTEND DOWNSTREAM A DISTANCE OF 10 EQUIVALENT
C     NOZZLE EXIT DIAMETERS (I.E. STA = 10 DIA), AND WILL CONTAIN
C     THE CUTS DEFINED IN THE FOLLOWING DATA STATEMENT.
C
      COMMON /DUMP/ IDMP(30)
      COMMON /RAXIN/ XNPR,EMJET,TTJET,THLIP,THETAJ,JDUM,IPRINT
     1               ,AMINF
      COMMON /PLUME/ MP(6),NP(6),XP(6,20,20),YP(6,20,20),
     1               ZP(6,20,20),VP(6,19,19)
      COMMON /CLINE/ XC,YC,ZC,A,B,C,EAREA,DIA
      COMMON /RAXOUT/ NPTS,XSTA(30),YRATIO(30),XVN(29)
      COMMON /XCUT/ NCUTS,CUT(11)
C
      DIMENSION STA(11),RATIO(11),VN(10),XSTAM(29),STAM(10)
      DATA CUT / 0.0,0.25,0.50,0.75,1.00,1.50,2.00,3.00,4.00,5.00,6.00/
C
      WRITE(11,*) ',NPTS= ',NPTS
C**************  CHECK DATA FROM RAXJET  ***************************
      IF(IDMP(6).EQ.1) WRITE(11,904)
 904  FORMAT(//,5X,'DUMP FROM SUBROUTINE EXTEND  ****************',
     1   '****************************',/,
     2   10X,'NONDIMENSIONAL OUTPUT FROM RAXJET FOLLOW',/,
     3   10X,' I     XSTA(I)    YRATIO(I)      XVN(I)')
      IF(IDMP(6).EQ.1) THEN
         DO 110  I=1,NPTS
            WRITE(11,906) I,XSTA(I),YRATIO(I)
 906        FORMAT(I10,2F10.3)
            IF(I.LT.NPTS) WRITE(11,908)  XVN(I)
 908        FORMAT(30X,F10.4)
 110     CONTINUE
      END IF
C
C
C**************  DEFINE THE CORNER POINTS OF THE PANELS  ***************
C
C
C    RATIO AT THE STATIONS IDENTIFIED IN THE XCUT DATA STATEMENT.
C
      NCUTS = 11
C
      DO I = 1,NCUTS
         STA(I) = CUT(I) * DIA
      END DO
C
```

```
      DO I = 1, NPTS
         XSTA(I) = XSTA(I) * DIA
      END DO
C
C
      CALL DER(XSTA,YRATIO,NPTS,SM,SN)
      CALL CAL(NPTS,XSTA,YRATIO,NCUTS,STA,RATIO,SM,SN,ISM)
C
C
C             EXTEND THE INITIAL POINTS ON THE PLUME DOWNSTREAM.
C
      DO 200  NET = 1,6
      IF(MP(NET).EQ.0) GO TO 200
C
      DO 120 I = 1, MP(NET)
         DO     J = 1,NCUTS
            XSLOPE = 1. - (STA(J) / STA(NCUTS))
            XP(NET,I,J) = XC + STA(J) +(XP(NET,1,1)-XC)*RATIO(J)*XSLOPE
            YP(NET,I,J) = YC + (YP(NET,I,1)-YC) * RATIO(J)
            DZ = A * STA(J)**2 + B * STA(J)
            ZP(NET,I,J) = ZC + DZ +(ZP(NET,I,1)-ZC) * RATIO(J)
C
C
         END DO
         NP(NET) = NCUTS
 120  CONTINUE
C
C
C
C
C***********  COMPUTE OUTFLOW VELOCITIES  *************************
C
C
C
      DO I = 1, NPTS
         XSTAM(I) = (XSTA(1) + XSTA(I+1)) / 2.0
      END DO
C
      DO I = 1,10
         STAM(I) = (STA(I) + STA(I+1)) / 2.0
      END DO
C
      N1 = NPTS -1
C
      CALL DER(XSTA,XVN,N1,SM,SN)
      CALL CAL(N1,XSTAM,XVN,10,STAM,VN,SM,SN,ISM)
C
C
C     FINAL VELOCITIES COMPUTED AT PANEL CENTERS.NOTE THERE IS
C     NO VN VARIATION IN THE M DIRECTION.
C
      DO 150 I = 1,MP(NET)
         DO J = 1,10
            VP(NET,I,J) = VN(J)
         END DO
 150  CONTINUE
C
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      IF(IDMP(6).EQ.1)WRITE(11,910)NET,((I,J,XP(NET,I,J),YP(NET,I,J),
     1    ZP(NET,I,J),VP(NET,I,J),I=1,MP(NET)), J= 1,NP(NET))
 910  FORMAT(//,5X,'COORDINATES FOR PLUME SEGMENT NUMBER ',I2,/,
     2   10X,'   M    N         XP           YP           ZP          VN',/,
     3       (10X,2I4,2X,4F10.3))
 200  CONTINUE
C
C
      RETURN
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE PBASE
C
C     SUBROUTINE PBASE GENERATES THE DATA FOR THE
C     BASE OF THE PLUME.  ONE PLUME BASE NETWORK FOR EACH
C     PLUME NETWORK.
C
      COMMON /DUMP/ IDMP(30)
      COMMON /PLUME/ MP(6),NP(6),XP(6,20,20),YP(6,20,20),
     1            ZP(6,20,20),VP(6,20,20)
      COMMON /CLINE/ XC,YC,ZC,Z,B,C,EAREA,DIA
      COMMON /XCUT/ NCUT,CUT(11)
C
      COMMON /PBASE/ MPB(6),NPB(6),XPB(6,20,2),YPB(6,20,2),
     1               ZPB(6,20,2)
C
      DXBASE = CUT(NCUT) * DIA
      DZ     = A * XBASE**2 + B * XBASE
C
      XBASE  = XC + DXBASE
      YBASE  = YC
      ZBASE  = ZC + DZ
C
      DO I = 1,6
C
C
C
         IF(MP(I).EQ.0) GO TO 100
         MPB(I) = MP(I)
         NPB(I) = 2
C
         DO J = 1, MP(I)
            XPB(I,J,1) = XP(I,J,NP(I))
            YPB(I,J,1) = YP(I,J,NP(I))
            ZPB(I,J,1) = ZP(I,J,NP(I))
C
            XPB(I,J,2) = XBASE
            YPB(I,J,2) = YBASE
            ZPB(I,J,2) = ZBASE
         END DO
 100  CONTINUE
C
C
      IF(IDMP(7).EQ.1.AND.MP(I).GT.0) WRITE(11,910) I,((J,K,XPB(I,J,K),
     1     YPB(I,J,K),ZPB(I,J,K), J=1,MPB(I)), K=1,NPB(I))
 910  FORMAT(//,5X,'DUMP FROM SUBROUTINE PBASE ********************',
     1     /,10X,'DATA FOLLOW FOR PLUME BASE SEGMENT NUMBER ',I2,/,
     2       10X,'  M     N       XPB       YPB       ZPB',
     3       /,(10X,2I4,3F10.3))
C
C
      END DO
C
      RETURN
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE WAKE
C
C     SUBROUTINE 'WAKE' DETERMINES THE COORDINATES FOR WAKE
C     NETWORKS THAT EMANATE FROM THE PLUME BASE.
C     EDGE 1 IS FORCED TO ABUT THE PLUME BASE.
C
C
      COMMON /DUMP/ IDMP(30)
      COMMON /CLINE/ XC,YC,ZC,A,B,C,EAREA,DIA
      COMMON /PBASE/ MPB(6),NPB(6),XPB(6,20,2),YPB(6,20,2),
     1               ZPB(6,20,2)
      COMMON /WAKE/ MW(6),NW(6),XW(6,2,20),YW(6,2,20),
     1               ZW(6,2,20)
C
C
C
C
C
C
C     MATE EDGE 1 OF WAKES TO THE PLUME BASE
      DO 100  I = 1, 6
      IF(MPB(I).EQ.0) GO TO 100
      NWAKE = I
C
      DO K = 1,MPB(I)
         XW(I,1,K) = XPB(1,K,1)
         YW(I,1,K) = YPB(I,K,1)
         ZW(I,1,K) = ZPB(I,K,1)
C
         XW(I,2,K) = XW(I,1,K) + 30. * DIA
         YW(I,2,K) = YW(1,1,K)
         ZW(I,2,K) = ZW(1,1,K)
C
         MW(I) = 2
         NW(I) = MPB(I)
C
      END DO
      IF(IDMP(8).EQ.1) WRITE(11,910) 1,((J,K,XW(1,J,K),
     1       YW(I,J,K),ZW(1,J,K),J=1,MW(I)),K=1,NW(1))
 910  FORMAT(/,/,5X,'DUMP FROM SUBROUTINE WAKE',
     1      '****************************************',/,
     2     13X,'THE COORDINATES FOLLOW FOR WAKE NUMBER ',I1,
     3     /,13X,'M    N       XW          YW         ZW',/,
     4   (10X,2I4,3F10.3))
 100  CONTINUE
      RETURN
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE CREATE
C
C     SUBROUTINE CREATE USES THE OLD PANAIR FILE AND THE
C     COMPUTED DATA TO CREATE A NEW PANAIR FILE WITH THE
C     PLUME NETWORKS AND THE ASSOCIATED OUTFLOW VELOCITIES.
C
      COMMON /TITL/   TITLE
      COMMON /DUMP/ IDMP(30)
      COMMON /PADAT/  OLDFIL, NEWFIL, ENAME,ABUT(6),IEDGE(6)
      COMMON /RAXIN/ XNPR,EMJET,TTJET,THLIP,THETAJ,JDUM,IPRINT
     1              ,AMINF
      COMMON /NETWORK/ NETWORK,M,N,X(20,20),Y(20,20),Z(20,20),
     1                BC,IFLAG
      COMMON /PLUME/ MP(6),NP(6),XP(6,20,20),YP(6,20,20),
     1               ZP(6,20,20),VP(6,19,19)
      COMMON /PBASE/ MPB(6),NPB(6),XPB(6,20,2),YPB(6,20,2),
     1               ZPB(6,20,2)
      COMMON /WAKE/ MW(6),NW(6),XW(6,2,20),YW(6,2,20),
     1              ZW(6,2,20)

C
C
      CHARACTER TITLE*80, OLDFIL*20, NEWFIL*20, LINE*80, ENAME*20,
     1          ABUT*20,NETWORK*20,BC*4,CARD*4
C
      REWIND 9
C
C****** READ AND WRITE UNTIL THE 'BEGIN GLOBAL DATA' CARD IS FOUND ****
C
 100  CONTINUE
      READ(9,110) LINE
 110  FORMAT(A80)
C
      DO I = 1, 20
         J = I + 3
         IF(LINE(I:J).EQ.'BEGI') GO TO 150
      END DO
      WRITE(10,120) LINE
 120  FORMAT(A80)
      GO TO 100
C
 150  CONTINUE
      DO I = 1, 20
         J = I + 3
         IF(LINE(I:J).EQ.'GLOB')  THEN
            WRITE(10,160)
            WRITE(10,162)
            WRITE(10,163)
            WRITE(10,164) TITLE(1:50), OLDFIL,NEWFIL
            IF(JDUM.NE.1) WRITE(10,166) XNPR,EMJET,TTJET,THLIP,THETAJ
            IF(JDUM.EQ.1) WRITE(10,168)
            WRITE(10,170)
         END IF
      END DO
C
 160  FORMAT(
     1 '*/    *****************************************************',
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
     2 '*****************',T74,'/ADDJET',/,
     3 '*/    ***************************************************',
     4 '*****************',T74,'/ADDJET',/,
     5 '*/    ***                                                 ',
     6 '                  ***',T74,'/ADDJET',/,
     7 '*/    ***              THE PLUME SIMULATION IN THIS PAN A',
     8 'IR                ***',T74,'/ADDJET',/,
     9 '*/    ***                 FILE WAS INCORPORATED WITH THE ',
     1 '                  ***',T74,'/ADDJET',/,
     1 '*/    ***                                                 ',
     2 '                  ***',T74,'/ADDJET',/,
     3 '*/    ***                     A D D J E T    C O D E    ',
     4 '                  ***',T74,'/ADDJET',/,
     5 '*/    ***                                                 ',
     6 '                  ***',T74,'/ADDJET',/,
     7 '*/    ***                          WRITTEN BY             ',
     8 '                  ***',T74,'/ADDJET')
C
C
 162  FORMAT(
     1 '*/    ***          GEORGE A. HOWELL AND ISHWAR C. BHATELE',
     2 'Y                 ***',T74,'/ADDJET',/,
     3 '*/    ***                GENERAL DYNAMICS CORPORATION  ',
     4 '                  ***',T74,'/ADDJET',/,
     5 '*/    ***                     FORT WORTH DIVISION       ',
     6 '                  ***',T74,'/ADDJET',/,
     7 '*/    ***                      FORT WORTH, TEXAS        ',
     8 '                  ***',T74,'/ADDJET',/,
     9 '*/    ***              PHONE (817) 732-4811 EXTENSION 44',
     1 '18                ***',T74,'/ADDJET',/,
     1 '*/    ***                                                 ',
     2 '                  ***',T74,'/ADDJET',/,
     3 '*/    ***                                                 ',
     4 '                  ***',T74,'/ADDJET')
 163  FORMAT(
     1 '*/    ***          THE 2-D JET CHARACTERISTICS WERE COMPU',
     2 'TED               ***',T74,'/ADDJET',/,
     3 '*/    ***        WITH THE RAXJET PLUME MODELING CODE DEVE',
     4 'LOPED             ***',T74,'/ADDJET',/,
     5 '*/    ***                    BY RICHARD G. WILMOTH      ',
     6 '                  ***',T74,'/ADDJET',/,
     7 '*/    ***               NASA/LANGLEY RESEARCH CENTER    ',
     7 '                  ***',T74,'/ADDJET',/,
     8 '*/    ***                                                 ',
     9 '                  ***',T74,'/ADDJET',/,
     1 '*/    ***                                                 ',
     2 '                  ***',T74,'/ADDJET')
C
 164  FORMAT(
     1 '*/    *** ',A50,T70,'*** /ADDJET',/,
     2 '*/    ***                                                 ',
     3 '                  ***',T74,'/ADDJET',/,
     4 '*/    ***            ORIGINAL FILE              =  ',A20,
     5               T70,'*** /ADDJET',/,
     6 '*/    ***            NEW FILE                   =  ',A20,
     7               T70,'*** /ADDJET')
C
```

```
 166  FORMAT(
     1 '*/    ***                                                    ',
     2              T70,'*** /ADDJET',/,
     5 '*/    ***              NOZZLE PRESSURE RATIO      =',F6.3,
     6              T70,'*** /ADDJET',/,
     7 '*/    ***              JET EXIT MACH NUMBER       =',F6.3,
     8              T70,'*** /ADDJET',/,
     9 '*/    ***              JET EXIT TEMPERATURE       =',F6.1,
     1              T70,'*** /ADDJET',/,
     1 '*/    ***              NOZZLE LIP ANGLE           =',F6.2,
     2              T70,'*** /ADDJET',/,
     3 '*/    ***              JET DEFLECTION ANGLE       =',F6.2,
     4              T70,'*** /ADDJET')
C
C
 168  FORMAT(
     1 '*/    ***                                                    ',
     2              T70,'*** /ADDJET',/.
     1 '*/    ***            A SOLID PLUME WITH NO INFLOW VELOCITY',
     2              T70,'*** /ADDJET',/,
     3 '*/    ***               IS SUMILATED IN THIS PAN AIR FILE ',
     4              T70,'*** /ADDJET')
C
 170  FORMAT(
     1 '*/    ***                                                    ',
     2              T70,'*** /ADDJET',/,
     3 '*/    ***                                                    ',
     4              T70,'*** /ADDJET',/,
     5 '*/    ***                                                    ',
     6              T70,'*** /ADDJET',/,
     1 '*/    ***********************************************',
     2 '****************',T74,'/ADDJET',/,
     3 '*/    ***********************************************',
     4 '****************',T74,'/ADDJET')
C
      WRITE (10,172)
 172  FORMAT('*/',/,'*/')
C
      WRITE(10,120) LINE
C
C****** READ AND WRITE NETWORK DATA UNTIL EXIT NETWORK IS FOUND ******
C
C
C
 200  CONTINUE
      NETWORK = 'XXXXXX'
      READ(9,110) LINE
      IF(LINE(1:2).EQ.'*/') GO TO 210
      DO I = 1, 20
         J = I + 3
         IF(LINE(I:J).EQ.'NETW') CALL READTTL(LINE,NETWORK)
         IF(NETWORK.EQ.ENAME) THEN
            CALL BLKEXIT(LINE,CARD)
            IF(CARD.EQ.'BEGI') GO TO 240
            WRITE(10,120) LINE
            GO TO 230
         END IF
```

```
      END DO
C
 210  CONTINUE
      WRITE(10,120) LINE
      GO TO 200
C
C*****  READ AND WRITE NETWORK DATA UNTIL NEXT 'BEGIN' CARD IS FOUND ***
C
 230  CONTINUE
      READ(9,110) LINE
      IF(LINE(1:2).EQ.'*/') GO TO 235
      DO I = 1,20
         J = I + 3
         IF(LINE(I:J).EQ.'BEGI') GO TO 240
      END DO
 235  CONTINUE
      WRITE(10,120) LINE
       GO TO 230
C
C
C
C******** INSERT PLUME, PLUME BASE, AND WAKE DATA  ********************
C
 240  CONTINUE
         CALL ADDPLU
         CALL ADDPLBA
         CALL ADDWAKE
C
C**********TRANSFER REMAINDER OF CARDS TO NEW FILE ****************
C
      WRITE(10,120) LINE
 260  CONTINUE
      READ(9,110,ERR=280) LINE
      WRITE(10,120) LINE
      GO TO 260
C
 280  CONTINUE
      WRITE(11,290)
 290  FORMAT(/,30X,'    PROGRAM COMPLETE')
      RETURN
      END
```

```
      SUBROUTINE READTIL(LINE,NETWORK)
C
C      SUBROUTINE 'READTIT' READS THE NETWORK TITLE CARD TO
C      DETERMINE THE NETWORK NAME.
C
C
      CHARACTER LINE*80, NETWORK*20
C
C*********** READ THE NETWORK TITLE  ************************
      DO I = 1,20
         IF(LINE(I:I).EQ.'=') GO TO 238
      END DO
C
C     NOW FIND START OF THE NETWORK NAME
 238  N1 = I + 1
      DO J = I+1 , 50
         IF(LINE(J:J).NE.' ') GO TO 240
         N1 = N1 + 1
      END DO
C
C
C     FIND END OF THE NETWORK NAME
 240  N2 = N1
      DO I = J, 50
         IF(LINE(I:I).EQ.' ') THEN
            N2 = I - 1
            GO TO 250
         END IF
      END DO
 250  NETWORK = LINE(N1:N2)
C
      RETURN
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE BLKEXIT(LINE,CARD)
C
C      SUBROUTINE "BLKEXIT" PLACES "*/" IN THE FIRST TWO
C      COLUMNS OF THE EXIT NETWORK CARDS TO NULLIFY
C     THE INPUT FOR THAT NETWORK.
C
      COMMON /DUMP/ IDMP(30)
C
       CHARACTER LINE*80, CARD*4
C
C
      WRITE(10,80)
 80   FORMAT('*/',T74,'/ADDJET',/,'*/',T74,'/ADDJET')
C
      WRITE(10,160)
 160  FORMAT(
     1'*/',10X,'**********************************************************',
     1       T74,'/ADDJET',/,'*/',
     2       10X,'**********************************************************',
     2       T74,'/ADDJET',/,'*/',
     3       10X,'**                                                      **',
     3       T74,'/ADDJET',/,'*/',
     4       10X,'**     THE FOLLOWING NETWORK HAS BEEN NULLIFIED         **',
     4       T74,'/ADDJET',/,'*/',
     6       10X,'**                  BY THE ADDJET CODE                  **',
     6       T74,'/ADDJET',/,'*/',
     6       10X,'**                                                      **',
     6       T74,'/ADDJET',/,'*/',
     7       10X,'**********************************************************',
     7       T74,'/ADDJET',/,'*/',
     8       10X,'**********************************************************',
     8       T74,'/ADDJET')
C
      WRITE(10,80)
      WRITE(10,90) LINE
 90   FORMAT('*/',A70,T74,'/ADDJET')
 100  CONTINUE
      READ(9,110) LINE
 110  FORMAT(A70)
      IF(LINE(1:2).EQ.'*/') GO TO 120
      DO I = 1,20
         J = I + 3
         CARD = LINE(I:J)
         IF(LINE(I:J).EQ.'NETW') GO TO 130
         IF(LINE(I:J).EQ.'BEGI') GO TO 130
      END DO
 120  CONTINUE
      WRITE(10,90) LINE
      GO TO 100
C
 130  CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ADDPLU
C
C     SUBROUTINE "ADDPLU" ADDS THE PLUME NETWORKS TO
C     THE NEW PAN AIR FILE.
C
      COMMON /DUMP/ IDMP(30)
      COMMON /PLUME/ MP(6),NP(6),XP(6,20,20),YP(6,20,20),
     1                    ZP(6,20,20),VP(6,19,19)
C
      DO 400 I = 1,6
      M = MP(I)
      N = NP(I)
      IF(M.EQ.0) GO TO 400
C
C**********  ADD A BANNER BEFORE THE PLUME DATA ***********
C
      WRITE(10,100)
  100 FORMAT('*/',T74,'/ADDJET',/,'*/',T74,'/ADDJET')
      IF(I.EQ.1) WRITE(10,130)
  130 FORMAT(
     1'*/',10X,'****************************************************',
     1        T74,'/ADDJET',/,'*/',
     2        10X,'****************************************************',
     2        T74,'/ADDJET',/,'*/',
     3        10X,'**                                                **',
     3        T74,'/ADDJET',/,'*/',
     4        10X,'**  THE FOLLOWING NETWORKS SIMULATE THE EXHAUST    **',
     4        T74,'/ADDJET',/,'*/',
     6        10X,'**        PLUME AND THE OUTFLOW VELOCITIES         **',
     6        T74,'/ADDJET',/,'*/',
     6        10X,'**                                                **',
     6        T74,'/ADDJET',/,'*/',
     7        10X,'****************************************************',
     7        T74,'/ADDJET',/,'*/',
     8        10X,'****************************************************',
     8        T74,'/ADDJET')
C
      WRITE(10,100)
C
C*********** WRITE NETWORK TITLE AND CORNER POINT DATA **************
C
      WRITE(10,135) I,M,N
  135 FORMAT('NETWORK=PLUME-SEGMENT-',I1,5X,I3,5X,I3,T74,'/ADDJET')
C
      DO K = 1,N
      DO J = 1, M
         WRITE(10,140) XP(I,J,K),YP(I,J,K),ZP(I,J,K)
  140     FORMAT(3F10.3,T74,'/ADDJET')
      END DO
      END DO
C
C
C*********** WRITE BOUNDARY CONDITION DATA, ETC.  *******************
C
      WRITE(10,150)
  150 FORMAT('TRIANGULAR PANEL TOLERANCE = .003',T74,'/ADDJET')
      WRITE(10,160)
```

```
 160  FORMAT('BOUNDARY CONDITION = 2,UPPER',T74,'/ADDJET')
      WRITE(10,170)
 170  FORMAT('SPECIFIED FLOW',T74,'/ADDJET')
      WRITE(10,180)
 180  FORMAT('TERM = 1' ,T74,'/ADDJET')
      WRITE(10,185)
 185  FORMAT('POINTS = ALL   $ 0.0',T74,'/ADDJET')
C
C
C*************** WRITE THE PLUME OUTFLOW VELOCITIES *****************
C
      WRITE(10,100)
      WRITE(10,190)
 190  FORMAT('*/        THE FOLLOWING DATA ARE THE OUTFLOW VELOCITIES'
     1      T74,'/ADDJET',/, '*/        THROUGH THE PANELS OF THE ',
     2        'PLUME NETWORK',T74,'/ADDJET')
      WRITE(10,200)
 200  FORMAT('POINTS = CENTER',T74,'/ADDJET')
C
      DO K = 1, N-1
         M1 = M - 1
         IF(M1.EQ.1) WRITE(10,1) (VP(I,J,K) ,J = 1, M-1)
         IF(M1.EQ.2) WRITE(10,2) (VP(I,J,K) ,J = 1, M-1)
         IF(M1.EQ.3) WRITE(10,3) (VP(I,J,K) ,J = 1, M-1)
         IF(M1.EQ.4) WRITE(10,4) (VP(I,J,K) ,J = 1, M-1)
         IF(M1.EQ.5) WRITE(10,5) (VP(I,J,K) ,J = 1, M-1)
         IF(M1.EQ.6) WRITE(10,6) (VP(I,J,K) ,J = 1, M-1)
         IF(M1.GE.7) WRITE(10,7) (VP(I,J,K) ,J = 1, M-1)
      WRITE(11,300) I,J,K, (VP(I,J,K),J= 1,M-1)
 300  FORMAT(5X,3I4,15F8.3)
      END DO
 1    FORMAT(1F10.3,T74,'/ADDJET')
 2    FORMAT(2F10.3,T74,'/ADDJET')
 3    FORMAT(3F10.3,T74,'/ADDJET')
 4    FORMAT(4F10.3,T74,'/ADDJET')
 5    FORMAT(5F10.3,T74,'/ADDJET')
 6    FORMAT(6F10.3,T74,'/ADDJET')
 7    FORMAT(7F10.3,T74,'/ADDJET')
C
      WRITE(10,100)
C
C
C
 400  CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ADDPLBA
C
C     SUBROUTINE 'ADDPLBA' ADDS THE PLUME BASE NETWORKS TO
C     THE NEW PAN AIR FILE.
C
      COMMON /DUMP/ IDMP(30)
      COMMON /PBASE/ MPB(6),NPB(6),XPB(6,20,2),YPB(6,20,2),
     1                ZPB(6,20,2)
C
      DO 400 I = 1,6
      M = MPB(I)
      N = NPB(I)
      IF(M.EQ.0) GO TO 400
C
C**********  ADD A BANNER BEFORE THE PLUME BASE DATA ************
C
      WRITE(10,100)
 100  FORMAT('*/',T74,'/ADDJET',/,'*/',T74,'/ADDJET')
      IF(I.EQ.1) WRITE(10,130)
 130  FORMAT(
     1'*/',10X,'*****************************************************',
     1       T74,'/ADDJET',/,'*/',
     2       10X,'*****************************************************',
     2       T74,'/ADDJET',/,'*/',
     3       10X,'**                                                 **',
     3       T74,'/ADDJET',/,'*/',
     4       10X,'**       THE FOLLOWING NETWORKS FORM THE BASE      **',
     4       T74,'/ADDJET',/,'*/',
     6       10X,'**                OF THE EXHAUST PLUME             **',
     6       T74,'/ADDJET',/,'*/',
     6       10X,'**                                                 **',
     6       T74,'/ADDJET',/,'*/',
     7       10X,'*****************************************************',
     7       T74,'/ADDJET',/,'*/',
     8       10X,'*****************************************************',
     8       T74,'/ADDJET')
C
      WRITE(10,100)
C
C*********** WRITE NETWORK TITLE AND CORNER POINT DATA **************
C
      WRITE(10,135) I,M,N
 135  FORMAT('NETWORK=PLUME-BASE-SEG-',I1,5X,I3,5X,I3,T74,
     1             '/ADDJET')
C
      DO K = 1,N
      DO J = 1, M
         WRITE(10,140) XPB(I,J,K),YPB(I,J,K),ZPB(I,J,K)
 140      FORMAT(3F10.3,T74,'/ADDJET')
      END DO
      END DO
C
C
C*********** WRITE BOUNDARY CONDITION DATA, ETC.  *******************
C
      WRITE(10,150)
 150  FORMAT('TRIANGULAR PANEL TOLERANCE = .003',40X,'/ADDJET')
```

```
      WRITE(10,160)
160   FORMAT('BOUNDARY CONDITION =  4,  8,1    6,3',T74,'/ADDJET')
      WRITE(10,170)
170   FORMAT('SINGULARITY TYPES =  SA, DA',T74,'/ADDJET')
400   CONTINUE
      RETURN
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE ADDWAKE
C
C     SUBROUTINE 'ADDWAKE' ADDS THE PLUME WAKE NETWORKS TO
C     THE NEW PAN AIR FILE.
C
      COMMON /DUMP/ IDMP(30)
      COMMON /WAKE/ MW(6),NW(6),XW(6,2,20),YW(6,2,20),
     1                  ZW(6,2,20)
C
      DO 400 I = 1,6
      M = MW(I)
      N = NW(I)
      IF(M.EQ.0) GO TO 400
C
C**********   ADD A BANNER BEFORE THE PLUME WAKE DATA ************
C
      WRITE(10,100)
 100  FORMAT('*/',T74,'/ADDJET',/,'*/',T74,'/ADDJET')
      IF(I.EQ.1) WRITE(10,130)
 130  FORMAT(
     1'*/',10X,'**************************************************',
     1      T74,'/ADDJET',/,'*/',
     2      10X,'**************************************************',
     2      T74,'/ADDJET',/,'*/',
     3      10X,'**                                              **',
     3      T74,'/ADDJET',/,'*/',
     4      10X,'**      THE FOLLOWING NETWORKS FORM THE WAKE     **',
     4      T74,'/ADDJET',/,'*/',
     6      10X,'**              FROM THE EXHAUST PLUME          **',
     6      T74,'/ADDJET',/,'*/',
     6      10X,'**                                              **',
     6      T74,'/ADDJET',/,'*/',
     7      10X,'**************************************************',
     7      T74,'/ADDJET',/,'*/',
     8      10X,'**************************************************',
     8      T74,'/ADDJET')
C
      WRITE(10,100)
C*********** WRITE NETWORK TITLE AND CORNER POINT DATA **************
C
      WRITE(10,135) I,M,N
 135  FORMAT('NETWORK=PLUME-WAKE-SEG-',I1,5X,I3,5X,I3,T74,
     1      '/ADDJET')
C
      DO K = 1,N
      DO J = 1, M
         WRITE(10,140) XW(I,J,K),YW(I,J,K),ZW(I,J,K)
 140       FORMAT(3F10.3,T74,'/ADDJET')
      END DO
      END DO
C
C
C*********** WRITE BOUNDARY CONDITION DATA, ETC.   ******************
C
      WRITE(10,150)
 150  FORMAT('WAKE FLOW PROPERTIES TAG',T74,'/ADDJET')
      WRITE(10,160)
```

```
160   FORMAT('BOUNDARY CONDITION =  1,  WAKE 1',T74,'/ADDJET')
400   CONTINUE
      RETURN
      END
```

```
***********************************************************************
C
C
***********************************************************************
C
C
***********************************************************************
C
      SUBROUTINE CAL(N,X,Y,NX,XB,YB,VM,VN,ISM)
C
C     SPLINE FIT  N VALUES OF Y AS A FUNCTION OF X
C
      COMMON /DUMP/ IDMP(30)
      DIMENSION X(1),Y(1),XB(1),YB(1)
      DIMENSION FP(30),FPP(30),FPPP(30),FIP(100)
C
      DATA PT/.33333333333333/
C
     IF(IDMP(7).EQ.1)  WRITE(11,910) (I,X(I),Y(I), I=1,N)
 910 FORMAT(///,20X, 'THE INPUT DATA TO CAL IS:',/,20X,
    1    ' I          X           Y',/,(16X,I4,2F10.3))
 6    CONTINUE
      ISM=0
      KM=1
      KN=1
      K = 1
      M = 1
      I = M
      J = M+K
      DS =X(J)-X(I)
      D = DS
      IF    (DS.EQ.0.) GO TO  150
      DF = (Y(J)-Y(I))/DS
      IF (KM-2) 10,20,30
 10   U = .5
      V = 3.*(DF-VM)/DS
      GO TO 50
 20   U = 0.
      V = VM
      GO TO 50
 30   U =  -1.
      V = -DS*VM
      GO TO 50
 40   I = J
      J= J+K
      DS = X(J)-X(I)
      IF (D*DS.LE.0.) GO TO 150
      DF = (Y(J)-Y(I))/DS
      B = 1./(DS+DS+U)
      U=B*DS
      V = B*(6.*DF-V)
 50   FP(I) = U
      FPP(I) = V
      U = (2.-U)*DS
      V = 6.*DF+DS*V
      IF (J.NE.N) GO TO 40
      IF(KN-2) 60,70,80
```

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

```
 60     V = (6.*VN-V)/U
        GO TO 90
 70     V = VN
        GO TO 90
 80     V = (DS*VN+FPP(I))/(1.+FP(I))
 90     B = V
        D = DS
 100    DS = X(J)-X(I)
        U = FPP(I)-FP(I)*V
        FPPP(I) = (V-U)/DS
        FPP(I) =  U
        FP(I) = (Y(J)-Y(I)) /DS -DS*(V+U+U)/6.
        V = U
        J = I
        I = I-K
        IF ( J.NE.M) GO TO 100
        FPPP(N) = FPPP(N-1)
        FPP(N) = B
        FP(N) = DF+D*(FPP(N-1)+B+B)/6.
        IF(NX.EQ.0) RETURN
C
C       INTERPOLATION FOR NX VALUES OF YB  AT XB
C
        J = 0
        DO 140 I=1,NX
        VAL = 0.
        VALP = 0.
        SS = XB(I)
 110    J = J+1
        TT = X(J)-SS
        IF(TT) 110,130,120
 120    J = J-1
        SS = SS-X(J)
        VAL = SS*(FP(J)+.5*SS*(FPP(J)+SS*PT*FPPP(J)))
        VALP = SS*(FPP(J)+.5*SS*FPPP(J))
 130    YB(I) =Y(J) + VAL
 140    FIP(I) = FP(J)+VALP
      IF(IDMP(7).EQ.1.) WRITE(11,920) (I,XB(I),YB(I), I=1,NX)
 920  FORMAT(///,20X,'THE OUTPUT DATA FROM CAL IS:',/20X,
     1      ' I       XB             YB',/,(16X,I4,2F10.3))
        RETURN
 150    CONTINUE
        WRITE(11,155)
 155    FORMAT(1X,'DATA COULD NOT BE SMOOTHED')
        ISM=1
        RETURN
 160    WRITE(11,170)
        ISM = 1
 170    FORMAT(/' SUBROUTINE SPLIF - - - NX=1'/)
        RETURN
        END
```

```
C**************************************************************
C
C
        SUBROUTINE DER(X,Y,N,VM,VN)
      COMMON /DUMP/ IDMP(30)
        DIMENSION X(1),Y(1)
C
C
       IF(IDMP(7).EQ.1) WRITE(11,100)
 100  FORMAT(//,'  INPUT TO DER ***',
     1     '    I      X(I)       Y(I)')
C
      IF(IDMP(7).EQ.1) WRITE(11,110) (I,X(I),Y(I), I=1,N)
 110  FORMAT(20X,I4,2F8.3)
        DX31 = X(3) - X(1)
        DX21 = X(2) - X(1)
        DX32 = X(3) - X(2)
        DY21 = Y(2) - Y(1)
        DY31 = Y(3) - Y(1)
        VM = DX31*DY21/DX21/DX32 - DX21*DY31/DX31/DX32
        DX31 = X(N-2) - X(N)
        DX21 = X(N-1) - X(N)
        DX32 = X(N-2) - X(N-1)
        DY21 = Y(N-1) - Y(N)
        DY31 = Y(N-2) - Y(N)
        VN = DX31*DY21/DX21/DX32 - DX21*DY31/DX31/DX32
        RETURN
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE RAXSUB
C
C      SUBROUTINE RAXJET IS A DUMMY SUBROUTINE THAT RETURNS
C     A SOLID, CIRCULAR PLUME TO THE ADDJET CODE.
C     THIS ALLOWS THE ADDJET CODE TO BE EXECUTED WITHOUT
C     RUNNING THE RAXJET CODE.
C
      COMMON /RAXIN/ XNPR,EMJET,TTJET,RJET,THLIP,THETAJ,JDUM,IPRINT
C
      COMMON /RAXOUT/ NPTS,XSTA(30),XRATIO(30),XVN(30)
C
      NPTS = 11
      XSTA(1) = 0.
      XSTA(2) = .5
      XSTA(3) = 1.
      XSTA(4) = 2.
      XSTA(5) = 3.
      XSTA(6) = 5.
      XSTA(7) = 7.
      XSTA(8) = 9.
      XSTA(9) = 10.
      XSTA(10)= 12.
      XSTA(11)=14.
C
      XRATIO(11)= 1.0
C
      DO   I= 1,NPTS
         XRATIO(I) = 1.0
         XVN(I) = 0.0
      END DO
C
C
      RETURN
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE RAXMAIN
C     PROGRAM RAXJET(INPUT,OUTPUT,TAPE1,TAPE2,TAPE3,TAPE4,TAPE5,TAPE6=OU
C    1TPUT,TAPE7,TAPE8)C   C   C   C   C   C   C   C   C   C   C   C
C
C     RAXJET - MAIN CONTROL PROGRAM
C
C
C
C************ COMMON BLOCK INPUT BY G.A. HOWELL **********************
C
      COMMON /HOWELL/ RSQGH(30), RSQDUM,IGH,REFL2
      COMMON /HOWELL2/ ISEN
      COMMON /RAXIN/ XNPR9,EMJET9,TTJET9,THLIP9,THETA9,JDUM,
     1                    IPRINT,AMINF9
C
C
C
C
      COMMON /CASE/ TITLE(20)
      COMMON /CNTRL/ IMAX,JMAX,MIT,NMIN,NMAX,ITERA,MVI,ISEP,IUI,IUO,NSEP
     1,IWAKE,IDPDX
      COMMON /FREE/ GAM,AMINF,PT,TT,REFL,REINF,SREF,PINF,ALPHAE(6),XS
      COMMON /GEOM/ IXY,IORDER,XO(150),YO(150),DYDXN
      COMMON /ROUT/ YB(81),XB(81),CP(81),FM(81)
      COMMON /SOLN/ P(81,42),DPO(81)
      COMMON /RLAX/ RF1,QF3,COVERG,WB,WBF,XOB(81),YOB(81)
      COMMON /XGRID/ XM,DXIDXO,DXIDXM,CXM,XBT
      COMMON /YGRID/ DNDYO,ALF,CT(81),AN(41)
      COMMON /BGRID/ IX1,IX2,JY1,JY2,IOFLG
      COMMON /BLOUT/ THR(81),DSTAR(81),DEL1(81),CFA(81),HS1(81),RET(81),
     1FNN(81),XSEP,XRE,CPO(81)
      COMMON /COUT/ IOFF,IAFT,IOUT,XSTRT,XEND,ROB,IMAP,JMAP,ALFO
      COMMON /DSL/ IS,IX,XDS(81),YDS(81),RDS(81),DELJ,RCMIN,DELE
      COMMON /JETCM/ XNPR,EMJET,TTJET,RJET,THLIP,NMAXJ,IGAS,ALPHAJ(6),GA
     1MJ,IDK
      COMMON /PEXTC/ IEXT,XEXT(30),PEXT(30),YPLU(30),PINT,XDK,UJET(30)
      COMMON /BINPT/ IVIS,MPSI,FDL,FFF,GGG,SIGMA,KMAXJ,KMAXE,ICONT,XJENT
     1,TKEJ,TKEX
      COMMON /BDSTAR/ BDUM(10)
      COMMON /BTOUT/ DELJO,XMAXB
C
C     IF MORE THAN 30 VISCOUS/INVISCID ITERATIONS ARE REQUIRED,
C        CHANGE THE DIMENSIONS IN THE FOLLOWING COMMON BLOCK
C         AND IN OVERLAY(1,0) AND OVERLAY(6,0)
C
      COMMON /CDRAG/ CDPAFT(30),CDFAFT(30),CDPBOD(30),CDFBOD(30)
      COMMON /IHIST/ FLAG(30),RPMAX(30),RPAVG(30),COVR
C
C
C************  7 OPEN STATEMENTS INSERTED BY G.A. HOWELL *************
C
      OPEN(UNIT=1,FILE='TAPE1.DAT',STATUS='UNKNOWN')
      OPEN(UNIT=2,FILE='TAPE2.DAT',STATUS='UNKNOWN')
      OPEN(UNIT=3,FILE='TAPE3.DAT',STATUS='UNKNOWN')
      OPEN(UNIT=4,FILE='TAPE4.DAT',STATUS='UNKNOWN')
      OPEN(UNIT=6,FILE='RAXJET.OUT',STATUS='UNKNOWN')
      OPEN(UNIT=7,FILE='TAPE7.DAT',STATUS='UNKNOWN')
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      OPEN(UNIT=8,FILE='TAPE8.DAT',STATUS='UNKNOWN')
C
      AMINF = AMINF9
C
C     CALL THE INPUT ROUTINE AND PRINT INITIAL DATA
C
      REWIND 8
C
C     CALL OVERLAY (4L$OBD,5,0)C     C     C     C     C     C     C     C
      CALL INPT
C
C     INITIAL CALL TO RAXBOD TO SET UP INVISCID GRID
C
      ITERA=0
C      CALL OVERLAY (4L$OBD,1,0)C      C      C      C      C      C      C
      CALL RAXBOD
      DO 20 I=2,IMAX
      IF (XB(I).GT.XB(I-1)) GO TO 20
      WRITE (6,270)
      STOP
20    CONTINUE
      IF (IOFF.EQ.0) GO TO 60
C
C     DEFINE GRID INDICES FOR OFFBODY CALCULATION
C
      IX1=1
      IX2=1
      JY1=JMAX
      JY2=2
      DO 40 I=2,IMAX
      IF (XSTRT.LT.XB(I)) GO TO 30
      IX1=I
30    IF (XEND.LE.XB(I)) GO TO 50
      IX2=I+1
40    CONTINUE
50    CONTINUE
C
C     SAVE OFFBODY PARAMETERS
C
      IX10=IX1
      IX20=IX2
      JY10=JY1
      JY20=JY2
      XSTRTO=XSTRT
      XENDO=XEND
      ROBO=ROB
60    CONTINUE
      IOFFO=IOFF
      IOFF=1
C
C     REDEFINE GRID INDICES FOR BOAT INVISCID MAP
C
      IX1=NMAX
      IX2=IMAX-1
      JY1=JMAX
      JY2=JMAX-KMAXE+1
      IF (JY2.LE.1) JY2=2
```

```
      ROB=YB(NMAX)+AN(JY2)
      XSTRT=XB(NMAX)
      XEND=XB(IX2)
C
C     DEFINE GRID INDEX AT START OF AFTERBODY FOR SEPARATION SEARCH
C
      NMIN=NMAX
      DO 70 I=1,NMAX
      IF (XBT.LT.XB(I)) GO TO 80
      NMIN=I
70    CONTINUE
C
C     SAVE ORIGINAL BODY GEOMETRY AND INITIALIZE AERODYNAMIC CONTOURS
C
80    IXY=IMAX-1
      DO 90 I=1,IXY
      XOB(I)=XB(I)
      YOB(I)=YB(I)
      XO(I)=XB(I)
      YO(I)=YB(I)
90    RDS(I)=YB(I)
C
C     INITIALIZE INVISCID JET PLUME BOUNDARY
C
      IEXT=IX2-IX1+1
      IF (IEXT.LE.30) GO TO 100
      WRITE (6,280)
      STOP
100   CONTINUE
      DO 110 I=1,IEXT
      N=I+NMAX-1
      XEXT(I)=XB(N)
110   YPLU(I)=RJET
      XSEP=XS
      IF (ISEP.NE.2) XSEP=XM
C
C     START VISCOUS/INVISCID LOOP
C
120   ITERA=ITERA+1
C
C     RECOMPUTE BODY SLOPE AT NOSE
C
      DX=XO(2)-XO(1)
      DY=YO(2)-YO(1)
      IF (DX.LE.DY*1.E-03) GO TO 130
      DYDXN=DY/DX
      GO TO 140
130   DYDXN=999.
140   CONTINUE
C
C     CALCULATE INVISCID EXTERNAL FLOW
C
C     CALL OVERLAY (4L$OBD,1,0)C     C     C     C     C     C     C     C
      CALL RAXBOD
C
C     INTERPOLATE AT NEW XB'S FOR YO, YOB, AND RDS
C
```

```
      CALL AINTPL (IXY,XB,XO,YO,IORDER)
      CALL AINTPL (IXY,XB,XO,RDS,IORDER)
      CALL AINTPL (IXY,XB,XOB,YOB,IORDER)
C
C     SET XO = XB, YO = YB, AND YB = YOB
C
      DO 150 I=1,IXY
      XOB(I)=XB(I)
      XO(I)=XB(I)
      YO(I)=YB(I)
150   YB(I)=YOB(I)
C
C     CALCULATE INVISCID JET EXHAUST
C
      IF (ITERA.GT.10.AND.XSEP.LT.XM) GO TO 160
C     CALL OVERLAY (4L$OBD,2,0)C    C    C    C    C    C    C    C
      CALL SCIPAC
160   CONTINUE
      I1=NMAX+1
      DO 170 I=I1,IXY
170   YB(I)=YPLU(I-I1+2)
C
C     CALCULATE BOUNDARY LAYER AND DISCRIMINATING STREAMLINE
C
      IMAX=IMAX-1
C     CALL OVERLAY (4L$OBD,3,0)C    C    C    C    C    C    C    C
      CALL VISCOUS
      IMAX=IMAX+1
C
C     CALCULATE MIXING LAYER AND PLUME DISPLACEMENT
C
C      CALL OVERLAY (4L$OBD,4,0)C     C     C     C     C     C     C
      CALL BOATAC
C
C     MAKE OFFBODY CALCULATION ON FINAL ITERATION
C
      IF (IOFFO.EQ.0) GO TO 200
      IF (ITERA.LT.MVI) GO TO 200
      IX1=IX10
      IX2=IX20
      JY1=JY10
      JY2=JY20
      XSTRT=XSTRTO
      XEND=XENDO
      ROB=ROBO
C
      DO 180 I=1,IXY
180   RDS(I)=YB(I)
C     CALL OVERLAY (4L$OBD,1,0)C    C    C    C    C    C    C    C
      CALL RAXBOD
      DO 190 I=1,IXY
190   YB(I)=RDS(I)
C
200   CONTINUE
      IF (XSEP.GE.XM) GO TO 220
      NR=IS+IX-2
      DO 210 I=IS,NR
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      IC=I-IS+2
210   YB(I)=YDS(IC)
220   CONTINUE
      IF (IOFFO.EQ.0.AND.ITERA.EQ.MVI) IOFF=0
C
C     OUTPUT RESULTS
C
C     CALL OVERLAY (4L$OBD,6,0)C    C    C    C    C    C    C    C
      CALL OUTPT
C
C     PERFORM UNDERRELAXATION TO GET NEW EFFECTIVE BODY
C
      W=WB
      IF (WB.EQ.WBF.OR.MVI.LE.1) GO TO 230
      W=WB+FLOAT(ITERA-1)/FLOAT(MVI-1)*(WBF-WB)
230   CONTINUE
      DO 240 I=1,IXY
      DELM1=YO(I)-RDS(I)
      RDS(I)=W*RDS(I)+(1.-W)*YB(I)
240   YO(I)=RDS(I)+W*DELM1+(1.-W)*DSTAR(I)
C
C     SMOOTH EFFECTIVE BODY GEOMETRY FOR SEPARATED FLOW
C
      IF (XSEP.GE.XM) GO TO 260
      NM1=IS-1
      IM1=IXY-1
      DO 250 N=NM1,IM1
      CALL SMOOTH (YO(N),YO(N-1),YO(N+1),XO(N),XO(N-1),XO(N+1))
250   CONTINUE
      YO(IXY)=YO(IM1)
260   CONTINUE
C
C     END OF VISCOUS/INVISCID LOOP
C
      IF (ITERA.LT.MVI) GO TO 120
C
C
C
C**************  2 LINES INPUT BY G.A. HOWELL  *********************
C
      REFL2 = REFL
C     CALL OVERLAY (4L$OBD,7,0)C    C    C    C    C    C    C    C
      CALL PANEL
      CLOSE(UNIT=1,DISPOSE='DELETE')
      CLOSE(UNIT=2,DISPOSE='DELETE')
      CLOSE(UNIT=3,DISPOSE='DELETE')
      CLOSE(UNIT=4,DISPOSE='DELETE')
      IF(IPRINT.EQ.0) CLOSE(UNIT=6,DISPOSE='DELETE')
      IF(IPRINT.EQ.1) CLOSE(UNIT=6,DISPOSE='KEEP')
      IF(IPRINT.EQ.2) CLOSE(UNIT=6,DISPOSE='PRINT/DELETE')
      IF(IPRINT.EQ.3) CLOSE(UNIT=6,DISPOSE='PRINT')
      CLOSE(UNIT=7,DISPOSE='DELETE')
      CLOSE(UNIT=8,DISPOSE='DELETE')
C
C
C
C
```

```
270   FORMAT (1H1,90H***  GRID PARAMETERS GIVE BAD TRANSFORMATION, TRY D
     1ECREASING CXM OR INCREASING DX1DXM  ***)
280   FORMAT (1H1,101H*** NUMBER OF AXIAL GRID POINTS IN JET EXHAUST REG
     1ION EXCEEDS MAXIMUM ALLOWED, TRY INCREASING CXM ***)
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE INPT
C
C     MAIN INPUT PROGRAM
C
C************ THREE COMMON BLOCKS ADDED BY G.A. HOWELL ****************
      COMMON /HOWELL2/ ISEN
      COMMON /RAXIN/ XNPR9,EMJET9,TTJET9,THLIP9,THETA9,JDUM,
     1                  IPRINT,AMINF9
      COMMON /RAXOUT/ NPTS,XSTA(30),XRATIO(30),XVN(30)
C
C
      COMMON /CASE/ TITLE(8)
      COMMON /CNTRL/ IMAX,JMAX,MIT,NMIN, MAX,ITERA,MVI,ISEP,IUI,IUO,NSEP
     1,IWAKE,IDPDX
      COMMON /FREE/ GAM,AMINF,PT,TT,REFL,REINF,SREF,PINF,ALPHAE(6),XSEP
      COMMON /GEOM/ IXY,IORDER,XO(150), O(150),DYDXN
      COMMON /RLAX/ RF1,QF3,COVERG,WB,WBF
      COMMON /XGRID/ XM,DXIDXO,DXIDXM,CXM,XBT
      COMMON /YGRID/ DNDYO,ALF
      COMMON /BGRID/ IX1,IX2,JY1,JY2
      COMMON /COUT/ IOFF,IAFT,IOU ,XSTRT,XEND,ROB,IMAP,JMAP,ALFO
      COMMON /JETCM/ XNPR,EMJET,  JET,RJET,THLIP,NMAXJ,IGAS,ALPHAJ(6),GA
     1MJ,IDK
      COMMON /BINPT/ IVIS,MPS1,FDL,FFF,GGG,SIGMA,KMAXJ,KMAXE,TCONT,XJENT
     1,TKEJ,TKEX
      DIMENSION S(150)
      NAMELIST /CNTRLN/ IMAX,JMAX,MIT,MVI,ISEP,IUI,IUO,NSEP,IWAKE,XSEP
     1,IDPDX
      NAMELIST /FSC/ GAM,AMINF,PT,TT,REFL,SREF,ALPHAE
      NAMELIST /GEOMN/ IXY,IORDER,XO,YO
      NAMELIST /RELAX/ RF1,QF3,COVERG,WB,WBF
      NAMELIST /GRID/ XBT,XM,DXIDXO,DXIDXM,CXM,DNDYO,ALF
      NAMELIST /OUTPTC/ IOFF,IAFT,IOUT,XSTRT,XEND,ROB,IMAP,JMAP,ALFO
      NAMELIST /JETDAT/ XNPR,EMJET,TTJET,RJET,THLIP,NMAXJ,IGAS,ALPHAJ,GA
     1MJ,IDK,ISEN
      NAMELIST /MIXDAT/ IVIS,MPS1,FDL,FFF,GGG,SIGMA,KMAXJ,KMAXE,TCONT,XJ
     1ENT,TKEJ,TKEX
C
C*********  ONE LINE INPUT BE G.A. HOWELL  *****************************
      CHARACTER TITLE*10
C
C
C     SET DEFAULT VALUES
C
      WRITE(11,5) XNPR9,EMJET9,TTJET9,THLIP9,THETA9,AMINF9
 5    FORMAT(//,' IN SUBROUTINE INPT, COMMON RAXIN = ',6F8.2,/)
C
      ISEN = 0
      IDPDX=0
      ID=81
      JD=41
      IXYD=140
      IS1=0
      IS2=0
      IMAX=ID
      JMAX=JD
      MIT=20
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      MVI=1
      ISEP=0
      NSEP=1
      IWAKE=0
      XSEP=-1.0
      IUI=0
      IUO=0
      GAM=1.4
      PT=-1.0
      TT=-1.0
      REFL=-1.0
      SREF=0.0
      IORDER=1
      RF1=1.4
      QF3=0.1
      COVERG=1.0
      WB=0.5
      WBF=0.5
      CXM=0.75
      ALF=1.3
      IOFF=0
      IAFT=0
      IOUT=0
      IMAP=25
      JMAP=25
      ALFO=1.0
      AMINF=0.0
      IXY=0
      DYDXN=0.0
      PI=3.14159265358898
      DO 20 I=1,IXYD
      XO(I)=-99.
20    YO(I)=-99.
      XBT=0.0
      XM=0.0
      DXIDXO=-1.
      DXIDXM=-1.
      DNDYO=-1.
      XSTRT=-1.
      XEND=-1.
      ROB=-1.
      XNPR=0.0
      EMJET=1.05
      TTJET=-1.0
      RJET=0.0
      THLIP=0.0
      NMAXJ=81
      IGAS=0
      GAMJ=0.0
      IDK=0
      ALPHAJ(1)=0.79
      ALPHAJ(2)=0.21
      ALPHAJ(3)=0.0
      ALPHAJ(4)=0.0
      ALPHAJ(5)=0.0
      ALPHAJ(6)=0.0
      IVIS=-2
```

```
      MPSI=31
      FDL=1.0
      FFF=0.065
      GGG=0.08
      TKEJ=1.0
      TKEX=1.0
      SIGMA=1.0
      KMAXJ=25
      KMAXE=25
      TCONT=-99.
      XJENT=15.
      ALPHAE(1)=0.79
      ALPHAE(2)=0.21
      ALPHAE(3)=0.0
      ALPHAE(4)=0.0
      ALPHAE(5)=0.0
      ALPHAE(6)=0.0
C
C
C**********THE FOLLOWING CARDS WERE MODIFIED OR ADDED BY G.A. HOWELL ***
C
C     READ CASE TITLE
C
C     READ (5,780,END=30) TITLE
C     GO TO 40
C     IF (EOF(5)) 30,40
C 30    WRITE (6,800)
C     STOP
C 40    CONTINUE
C
C     READ INPUT DATA
C
C     READ (5,CNTRLN)
C     READ (5,FSC)
C     READ (5,JETDAT)
C     READ (5,MIXDAT)
C     READ (5,RELAX)
C     READ (5,GRID)
C     READ (5,OUTPTC)
C     READ (5,GEOMN)
C
      CALL INPT2
      WRITE(11,45) XNPR,EMJET,TTJET,THLIP,AMINF
 45   FORMAT(//, ' IN SUBROUTINE INPT AND AFTER INPT2, THE VALUES IN',
     1   ' COMMON /RAXIN/ ARE:',/, '    XNPR,    EMJET,    TTJET,    THLIP',
     2          '       AMINF',/,5F9.2,/)
      WRITE(11,46) IXY,RJET,XSTRT,XEND,ROB,(I,XO(I),YO(I),I = 1,10)
 46   FORMAT(/,' IXY = ',I8,/,' RJET = ',F10.3,/,' XSTRT = ',F10.3,/,
     1 ' XEND = ',F10.3,/,' ROB  = ',F10.3,/,'     I     XO(I)     YO(I)',/
     2        (I5,2F8.2))
C
C
C*********************************************************************
C
C
C
      CALL DATE (DAY)
```

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

```
      CALL CLOCK (TIME)
      INAME=0
50    INAME=INAME+1
      WRITE (6,1430)
      WRITE (6,1390)
      WRITE (6,1400)
      WRITE (6,1410) TITLE
      WRITE (6,1420) DAY,TIME
      IF (INAME.EQ.1) GO TO 50
C
C     CONVERT INPUT TO INTERNAL UNITS
C
      IF (IUI.EQ.0) GO TO 60
      REFL=REFL/0.3048
      SREF=SREF/0.3048**2
      PT=PT*2116.

      TT=TT*1.8
      TTJET=TTJET*1.8
      TCONT=TCONT*1.8
60    IF (REFL.LE.0.0) REFL=1.0
      IF (PT.LE.0.0) PT=2116.
      IF (TT.LE.0.0) TT=530.
      IF (TTJET.LE.0.0) TTJET=530.
CCC         TTJET = TTJET/1.8
      TCONT=TCONT/1.8
C
C     CHECK FOR INVALID INPUT
C
      WRITE (6,790)
      IF (IOUT.GT.MVI) IOUT=MVI
      IF (SREF.EQ.0.0) SREF=0.25*PI*REFL**2
      IF (REFL.GT.0.0) GO TO 70
      WRITE (6,810)
      IS1=1
70    IF (SREF.GT.0.0) GO TO 80
      WRITE (6,820)
      IS1=1
80    IF (AMINF.GT.0.0) GO TO 90
      WRITE (6,830)
      IS1=1
90    IF (IXY.GT.3) GO TO 100
      WRITE (6,840)
      IS1=1
100   IF (XBT.GT.0.0) GO TO 110
      WRITE (6,850)
      IS1=1
110   IF (XM.GT.0.0) GO TO 120
      WRITE (6,860)
      IS1=1
120   IF (XBT.LE.XM) GO TO 130
      WRITE (6,870)
      IS1=1
130   CONTINUE
      IF (IXY.GT.IXYD) GO TO 260
      ICX=0
      ICY=0
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      DO 140 I=1,IXYD
      IF (XO(I).EQ.-99.) ICX=ICX+1
      IF (YO(I).EQ.-99.) ICY=ICY+1
140   CONTINUE
      ICX=IXYD-ICX
      ICY=IXYD-ICY
      IF (ICX.EQ.IXY.AND.ICY.EQ.IXY) GO TO 150
      IS1=1
      IF (ICX.NE.IXY) WRITE (6,910)
      IF (ICY.NE.IXY) WRITE (6,920)
      GO TO 260
150   CONTINUE
      DO 160 I=2,IXY
      IF (XO(I).GT.XO(I-1)) GO TO 160
      WRITE (6,930) I,XO(I)
      IS1=1
160   CONTINUE
      DO 170 I=1,IXY
      I1=I
      IF (XO(I).LE.XM) GO TO 170
      I1=I-1
      XM=XO(I1)
      WRITE (6,940) I1,XM
      GO TO 180
170   CONTINUE
180   IXY=I1
      DO 190 I=2,IXY
      IF (YO(I).GT.YO(I-1)) GO TO 190
      I1=I
      GO TO 200
190   CONTINUE
200   CONTINUE
      YMIN=0.1*YO(I1-1)
      DO 220 I=I1,IXY
      DY=YO(I)-YO(I-1)
      DX=XO(I)-XO(I-1)
      IF (DX.GT.0.0) GO TO 210
      WRITE (6,960) XO(I)
      IS1=1
      GO TO 220
210   DYDX=DY/DX
      IF (ABS(DYDX).GT.1.0) WRITE (6,950) I,XO(I)
      IF (YO(I).GE.YMIN) GO TO 220
      WRITE (6,970)
      IS1=1
220   CONTINUE
      IF (IOFF.EQ.0) GO TO 260
      IF (IMAP+JMAP.LE.100) GO TO 230
      IMAP=50
      JMAP=50
      WRITE (6,890)
230   IF (ALFO.GT.0.0) GO TO 240
      WRITE (6,900)
      ALFO=1.0
240   CONTINUE
      IF (XSTRT.GT.XEND) GO TO 250
      IF (XSTRT.GE.0.0.AND.XEND.GT.0.0.AND.ROB.GT.0.0) GO TO 260
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
250   WRITE (6,880)
      IOFF=0
260   IF (IMAX.LE.ID.AND.JMAX.LE.JD) GO TO 270
      WRITE (6,980) ID,JD
      IF (IMAX.GT.ID) IMAX=ID
      IF (JMAX.GT.JD) JMAX=JD
270   IF (GAM.GT.1.0) GO TO 280
      WRITE (6,990)
      IS1=1
280   IF (GAM.LT.2.0) GO TO 290
      WRITE (6,1000) GAM
      IS2=1
290   CONTINUE
      IF (PT.GT.0.0) GO TO 300
      WRITE (6,1010)
      IS1=1
300   IF (TT.GT.0.0) GO TO 310
      WRITE (6,1020)
      IS1=1
310   IF (ISEP.NE.2) GO TO 320
      IF (XSEP.GE.XBT.AND.XSEP.LE.XM) GO TO 320
      WRITE (6,1030)
      IS1=1
320   IF (IXY.LE.IXYD) GO TO 330
      WRITE (6,1040)
      IS1=1
330   IF (IORDER.GE.1) GO TO 340
      WRITE (6,1050)
      IORDER=1
      IS2=1
340   IF (IORDER.LE.2) GO TO 350
      WRITE (6,1060)
      IORDER=2
      IS2=1
350   DY=YO(2)-YO(1)
      DX=XO(2)-XO(1)
      IF (DX.GT.0.0.AND.DY.GT.0.0) GO TO 360
      WRITE (6,1070)
      IS1=1
      GO TO 370
360   DYDXN=DY/DX
      IF (DYDXN.GE.1.E03) DYDXN=999.
370   IF (RF1.GT.0.0.AND.RF1.LT.2.0) GO TO 380
      WRITE (6,1080)
      IS1=1
380   IF (QF3.GE.0.0) GO TO 390
      WRITE (6,1090)
      IS1=1
390   IF (WB.GE.0.0.AND.WB.LE.1.0) GO TO 400
      WRITE (6,1100)
      IS1=1
400   IF (WBF.GE.0.0.AND.WBF.LE.1.0) GO TO 410
      WRITE (6,1110)
      IS1=1
410   IF (CXM.GT.0.0.AND.CXM.LT.1.0) GO TO 420
      WRITE (6,1120)
      CXM=0.75
```

```
      IS2=1
420   IF (ALF.GT.0.0) GO TO 430
      WRITE (6,1130)
      ALF=1.3
      IS2=1
430   NMAX=INT(CXM*FLOAT(IMAX))+1
      IF (NMAX.GE.4) GO TO 440
      WRITE (6,1140)
      IS1=1
440   IF (DXIDXO.GT.0.0) GO TO 450
      WRITE (6,1150)
      DXIDXO=0.0
      IS2=2
450   IF (DXIDXM.GT.0.0) GO TO 460
      WRITE (6,1160)
      DXIDXM=0.0
      IF (IS2.LT.2) IS2=3
      IF (IS2.EQ.2) IS2=4
460   IF (DNDYO.GT.0.0) GO TO 470
      WRITE (6,1170)
      DNDYO=0.0
      IF (IS2.LE.1) IS2=5
      IF (IS2.EQ.2) IS2=6
      IF (IS2.EQ.3) IS2=7
      IF (IS2.EQ.4) IS2=8
470   CONTINUE
      IF (XNPR.GT.0.0) GO TO 480
      WRITE (6,1210)
      IS1=1
480   IF (EMJET.GT.1.0) GO TO 490
      WRITE (6,1220)
      IS1=1
490   IF (TTJET.GT.0.0) GO TO 500
      WRITE (6,1230)
      IS1=1
500   IF (RJET.GT.0.0) GO TO 510
      WRITE (6,1240)
      IS1=1
      GO TO 530
510   IF (RJET.LE.YO(IXY)) GO TO 520
      WRITE (6,1250)
      IS1=1
      GO TO 530
520   IF (RJET.EQ.YO(IXY)) GO TO 530
      DUM=YO(IXY)-RJET
      WRITE (6,1260) DUM
530   IF (ABS(THLIP).LT.90.) GO TO 540
      WRITE (6,1270)
      IS1=1
540   IF (NMAXJ.LE.81) GO TO 550
      NMAXJ=81
      WRITE (6,1280) NMAXJ
550   IF (IGAS.GE.0.AND.IGAS.LE.3) GO TO 560
      WRITE (6,1290)
      IS1=1
560   IF (IGAS.EQ.0) GAMJ=1.4
      IF (IGAS.GT.1) GO TO 570
```

```
      IF (GAMJ.GT.1.0) GO TO 570
      WRITE (6,1300)
      IS1=1
570   SUM=0.0
      DO 580 I=1,6
580   SUM=SUM+ALPHAJ(I)
      IF (SUM.LE.1.01.AND.SUM.GE.0.99) GO TO 590
      WRITE (6,1310)
      IS1=1
590   CONTINUE
C
C     EXTEND INPUT COORDINATES OVER JET EXHAUST
C
      DO 600 I=1,10
      N=IXY+I
      XO(N)=XO(IXY)+FLOAT(I)*RJET
600   YO(N)=RJET
      IXY=IXY+10
      IF (IVIS.EQ.2.OR.IVIS.EQ.0.OR.IVIS.EQ.-2) GO TO 610
      WRITE (6,1320) IVIS
      IVIS=-2
610   IF (MPSI.LE.50) GO TO 620
      WRITE (6,1330) MPSI
      MPSI=50
620   IF (FDL.LE.1.0) GO TO 630
      WRITE (6,1340) FDL
      FDL=1.0
630   IF (KMAXJ.LE.25) GO TO 640
      WRITE (6,1350) KMAXJ
      KMAXJ=25
640   IF (KMAXE.LE.25) GO TO 650
      WRITE (6,1360) KMAXE
      KMAXE=25
650   IF (TCONT.LE.0.) TCONT=400./FLOAT(MPSI-1)
      IF (XJENT.GE.2.0.AND.XJENT.LE.30.0) GO TO 660
      WRITE (6,1370) XJENT
660   SUM=0.0
      DO 670 I=1,6
670   SUM=SUM+ALPHAE(I)
      IF (SUM.LE.1.01.AND.SUM.GE.0.99) GO TO 680
      WRITE (6,1380)
      IS1=1
680   CONTINUE
      IF (IS1.GT.0) GO TO 760
      IF (IS2.EQ.0) GO TO 750
      IF (IS2.EQ.1) GO TO 740
C
C     CALCULATE GRID PARAMETERS IF NEEDED
C
      S(1)=0.0
      IH=IXY
      DO 720 I=2,IXY
      DY=YO(I)-YO(I-1)
      IF (I.LT.IH) GO TO 690
      DY=0.0
      GO TO 700
690   IF (DY.LE.0.0) IH=I-1
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
700   S(I)=S(I-1)+SQRT((XO(I)-XO(I-1))**2+DY**2)
      IF (XO(I).GT.XB1) GO TO 710
      S1=S(I)
710   IF (XO(I).GT.XM) GO TO 730
      S2=S(I)
720   CONTINUE
730   CALL OPTIM (IS2,IMAX,JMAX,S1,S2,CXM,DX1DXO,DX1DXM,DNDYO,ALF)
      NMAX=INT(CXM*FLOAT(IMAX-1))+1
      GO TO 770
740   WRITE (6,1200)
      GO TO 770
750   WRITE (6,1190)
      GO TO 770
760   WRITE (6,1180)
      STOP
770   CONTINUE
      CALL IOUTPT
C
780   FORMAT (8A10)
790   FORMAT (1H1,23H ---WARNING MESSAGES---,//)
800   FORMAT (1H1,38H END-OF-FILE ENCOUNTERED ON INPUT FILE)
810   FORMAT (49H   /REFERENCE LENGTH ZERO OR NEGATIVE, TRY AGAIN/)
820   FORMAT (47H   /REFERENCE AREA ZERO OR NEGATIVE, TRY AGAIN/)
830   FORMAT (57H   /FREESTREAM MACH NO. NOT GIVEN OR NEGATIVE, TRY AGAI
     1N/)
840   FORMAT (90H   /NO. OF INPUT COORDINATES, IXY, EITHER NOT GIVEN OR
     1TOO SMALL (MUST BE > 3), TRY AGAIN/)
850   FORMAT (55H   /START OF BOATTAIL NOT GIVEN OR NEGATIVE, TRY AGAIN/
     1)
860   FORMAT (49H   /END OF BODY NOT GIVEN OR NEGATIVE, TRY AGAIN/)
870   FORMAT (59H   /START OF AFTERBODY GREATER THAN BODY LENGTH, TRY AG
     1AIN/)
880   FORMAT (80H   OFFBODY COORDINATES NOT GIVEN OR NOT PROPER, NO OFFB
     1ODY CALCULATION ATTEMPTED)
890   FORMAT (86H   TOTAL MAPPED POINTS FOR OFFBODY FLOWFIELD EXCEEDS 10
     10, IMAP,JMAP CHANGED TO 50 EACH)
900   FORMAT (40H   ALFO MUST BE > 0, ALFO CHANGED TO 1.0)
910   FORMAT (50H   /NO. OF INPUT X-COORDINATES DOES NOT EQUAL IXY/)
920   FORMAT (50H   /NO. OF INPUT Y-COORDINATES DOES NOT EQUAL IXY/)
930   FORMAT (31H   /BAD VALUE IN X-ARRAY AT I =,I5,6H, X = ,F12.4,12H,
     1TRY AGAIN/)
940   FORMAT (53H   INPUT COORDINATES GIVEN BEYOND SPECIFIED XM, ONLY ,I
     15,37H VALUES WILL BE USED AND XM RESET TO ,F12.6)
950   FORMAT (34H   BODY SLOPE > 45 DEGREES AT I = ,I5,6H, X = ,F12.6,36
     1H, UNRELIABLE RESULTS MAY BE PRODUCED)
960   FORMAT (33H   /BAD VALUE OF X-COORDINATE AT ,F12.6,34H GIVES NEGAT
     1IVE OR INFINITE SLOPE/)
970   FORMAT (57H   /AFTERBODY CLOSURE > 90 % OF MAX. DIAMETER, TRY AGAI
     1N/)
980   FORMAT (32H   GRID SIZE EXCEEDS MAXIMUM OF ,I5,4H BY ,I5,54H ALLOW
     1ED, IMAX AND/OR JMAX REDUCED TO MAXIMUM VALUE(S))
990   FORMAT (43H   /INPUT VALUE OF GAM .LE. 1.0, TRY AGAIN/)
1000  FORMAT (46H   INPUT VALUE OF GAM .GE. 2.0, IS THIS RIGHT?)
1010  FORMAT (40H   /INPUT VALUE OF PT .LE. 0, TRY AGAIN/)
1020  FORMAT (40H   /INPUT VALUE OF TT .LE. 0, TRY AGAIN/)
1030  FORMAT (60H   /XSEP NOT GIVEN OR NOT ON BOATTAIL FOR ISEP=2, TRY A
     1GAIN/)
```

```
1040  FORMAT (42H   /INPUT VALUE OF IXY EXCEEDS MAXIMUM OF ,I5,12H, TRY
     1AGAIN/)
1050  FORMAT (60H   INPUT VALUE OF IORDER < 1 NOT ALLOWED, IORDER = 1 AS
     1SUMED)
1060  FORMAT (60H   INPUT VALUE OF IORDER > 2 NOT ALLOWED, IORDER = 2 AS
     1SUMED)
1070  FORMAT (55H   /INPUT COORDINATES AT NOSE GIVE BAD VALUE FOR DYDXN/
     1)
1080  FORMAT (58H   /INPUT VALUE OF RF1 NOT IN THE RANGE 0 TO 2, TRY AGA
     1IN/)
1090  FORMAT (43H   /INPUT VALUE OF QF3 NEGATIVE, TRY AGAIN/)
1100  FORMAT (57H   /INPUT VALUE OF WB NOT IN THE RANGE 0 TO 1, TRY AGAI
     1N/)
1110  FORMAT (58H   /INPUT VALUE OF WBF NOT IN THE RANGE 0 TO 1, TRY AGA
     1IN/)
1120  FORMAT (70H   INPUT VALUE OF CXM NOT IN THE RANGE 0 TO 1, A VALUE
     1OF 0.75 ASSUMED)
1130  FORMAT (52H   INPUT VALUE OF ALF .LE. 0, A VALUE OF 1.3 ASSUMED)
1140  FORMAT (77H   /INPUT VALUE OF CXM GIVES < 4 GRID POINTS ON THE BOD
     1Y, TRY INCREASING CXM/)
1150  FORMAT (81H   INPUT VALUE OF DXIDXO NOT GIVEN OR < 0, WILL TRY TO
     1CALCULATE AN OPTIMUM VALUE)
1160  FORMAT (81H   INPUT VALUE OF DXIDXM NOT GIVEN OR < 0, WILL TRY TO
     1CALCULATE AN OPTIMUM VALUE)
1170  FORMAT (81H   INPUT VALUE OF DNDYO  NOT GIVEN OR < 0, WILL TRY TO
     1CALCULATE AN OPTIMUM VALUE)
1180  FORMAT (///,62H ---INPUT ERRORS ENCLOSED BY /  / ARE FATAL, RUN IS
     1 ABORTED---)
1190  FORMAT (12H0 ---NONE---)
1200  FORMAT (///,82H ---IF ANY INPUT PARAMETERS WERE CHANGED, RUN WILL
     1BE ATTEMPTED WITH NEW VALUES---)
1210  FORMAT (59H   /NOZZLE PRESSURE RATIO NOT GIVEN OR NEGATIVE, TRY AG
     1AIN/)
1220  FORMAT (45H   /INPUT VALUE OF EMJET .LE. 1.0, TRY AGAIN/)
1230  FORMAT (43H   /INPUT VALUE OF TTJET .LE. 0, TRY AGAIN/)
1240  FORMAT (39H   /INPUT VALUE OF RJET = 0, TRY AGAIN/)
1250  FORMAT (45H   /INPUT VALUE OF RJET > YO(IXY), TRY AGAIN/)
1260  FORMAT (49H   NOZZLE GEOMETRY GIVEN HAS A BASE THICKNESS OF ,F12.6
     1,46H, TOO LARGE A BASE MAY GIVE UNRELIABLE RESULTS)
1270  FORMAT (49H   /NOZZLE EXIT ANGLE .GE. 90 DEGREES, TRY AGAIN/)
1280  FORMAT (52H   JET GRID SIZE EXCEEDS MAXIMUM ALLOWED, REDUCED TO,I5
     1)
1290  FORMAT (63H   /INCORRECT VALUE FOR IGAS, MUST BE 0, 1, 2, OR 3, TR
     1Y AGAIN/)
1300  FORMAT (40H   /GAMJ NOT INPUT OR .LE. 1, TRY AGAIN/)
1310  FORMAT (65H   /SUM OF JET MOLE FRACTIONS NOT WITHIN 1 % OF UNITY,
     1TRY AGAIN/)
1320  FORMAT (47H   IVIS MUST BE -2, 0, OR 2, IVIS CHANGED FROM ,I5,6H T
     1O -2)
1330  FORMAT (43H   MPSI MUST BE .LE. 50, MPSI CHANGED FROM ,I5,6H TO 50
     1)
1340  FORMAT (40H   FDL MUST BE .LE. 1, FDL CHANGED FROM ,F12.6,7H TO 1.
     10)
1350  FORMAT (45H   KMAXJ MUST BE .LE. 25, KMAXJ CHANGED FROM ,I5,6H TO
     125)
1360  FORMAT (45H   KMAXE MUST BE .LE. 25, KMAXE CHANGED FROM ,I5,6H TO
     125)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
1370  FORMAT (35H   MIXING CALCULATION EXTENDS OVER ,F12.6,40H JET RADII
     1.  CHECK INPUT VALUE OF XJENT.)
1380  FORMAT (70H   /SUM OF EXTERNAL MOLE FRACTIONS NOT WITHIN 1 % OF UN
     1ITY, TRY AGAIN/)
1390  FORMAT (15(/),20X,71HRRRRRRRRR       AAAAAA     XXX     XXX          JJ
     1J  EEEEEEEEEE  TTTTTTTTTTT,/,20X,71HRRRRRRRRR     AAAAAAAA    XXX
     2 XXX           JJJ  EEEEEEEEEE  TTTTTTTTTTT,/,20X,67HRRRR     RRR  AAA
     3     AAA    XXX   XXX            JJJ   EEE                TTT,/,20X,67HRRRR
     4    RRR  AAA     AAA    XXX   XXX              JJJ   EEE              TTT,/,
     520X,67HRRRR     RRR  AAA     AAA     XXXXXX              JJJ   EEE
     6       TTT,/,20X,67HRRRRRRRRRR  AAAAAAAAAA      XXXX              JJJ
     7EEEEEEEE         TTT,/,20X,67HRRRRRRRRR   AAAAAAAAAA       XX
     8       JJJ  EEEEEEEE         TTT)
1400  FORMAT (20X,67HRRRRRR         AAA     AAA      XXXX               JJJ  EEE
     1               TTT,/,20X,67HRRRRRRR        AAA     AAA     XXXXXX
     2    JJJ  EEE               TTT,/,20X,67HRRRR RRR      AAA     AAA    XXX
     3 XXX           JJJ  EEE                TTT,/,20X,67HRRRR  RRR     AAA
     4 AAA    XXX   XXX    JJJ     JJJ  EEE              TTT,/,20X,67HRRRR    R
     5RR    AAA     AAA   XXX     XXX    JJJJJJ,,'  EEEEEEEEEE       TTT,/,20X
     6,67HRRRR     RRR  AAA     AAA   XXX      XXX     JJJJJJ     EEEEEEEEEE
     7  TTT)
1410  FORMAT (/////,20X,7HCASE - ,8A10)
1420  FORMAT (/////,34X,6HDATE -,A10,10X,6HTIME -,A10)
1430  FORMAT (1H1)
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE INPT2
C
C
C     THIS SUBROUTINE IS CALLED BY SUBROUTINE INPT IN THE
C     RAXJET CODE.  IT'S PURPOSE IS TO DEFINE DEFAULT VALUES
C     OF CERTAIN PARAMETERS DIFFERENTLY THAN THOSE DEFINED
C     BY THE ORGINAL RAXJET CODE.
C
      COMMON /RAXIN/ XNPR9,EMJET9,TTJET9,THLIP9,THETA9,JDUM,
     1                  IPRINT,AMINF9
C
      COMMON /RAXOUT/ NPTS,XSTA(30),XRATIO(30),XVN(30)
C
C     NAMELIST (FSC)
      COMMON /FREE/ GAM,AMINF,PT,TT,REFL,REINF,SREF,PINF,ALPHAE(6),XSEP
C
C     NAMELIST (CNTRLN)
      COMMON /CNTRL/ IMAX,JMAX,MIT,NMIN,NMAX,ITERA,MVI,
     1                  ISEP,IUI,IUO,NSEP,IWAKE,IDPDX
C     NAMELIST (JETDAT)
      COMMON /JETCM/ XNPR,EMJET,TTJET,RJET,THLIP,NMAXJ,IGAS,
     1                  ALPHAJ(6),GAMJ,IDK
C      NAMELIST (MIXDAT)
      COMMON /BINPT/ IVIS,MPSI,FDL,FFF,GGG,SIGMA,KMAXJ,KMAXE,ICONT,
     1                  XJENT,TKEJ,TKEX
C     NAMELIST(OUTPIC)
      COMMON /COUT/ IOFF,IAFT,IOUT,XSTRT,XEND,ROB,IMAP,JMAP,ALFO
C     NAMELIST (GEOMN)
      COMMON /GEOM/ IXY,IORDER,XO(150),YO(150),DYDXN
      COMMON /XGRID/ XM,DXIDXO,DXIDXM,CXM,XBT
C
      COMMON /TITL/ TITLE1
      COMMON /CASE/ TITLE(8)
      CHARACTER TITLE*10, TITLE1*80
C
      DIMENSION XP(8),YP(8)
      DATA XP /0.,10.,20.,30.,40.,50.,60.,70./
      DATA YP /0.,5.,10.,10.,10.,10.,10.,7.2/
C
C
C     SET THE RAXJET TITLE EQUAL TO THE TITLE READ IN ADDJET
      DO  I =1,8
         J = I * 10 - 9
         JJ= I * 10
         TITLE(I) = TITLE1(J:JJ)
      END DO
C
      XNPR = XNPR9
      EMJET = EMJET9
      TTJET = TTJET9
      THLIP = THLIP9
      THETAJ = THETA9
      AMINF = AMINF9
C
C
      IMAX  = 51
      JMAX =  11
```

```
      MIT =  10
      IDPDX =  1
      NMAXJ =  41
      IGAS =  2
      IDK = 2
      XJENT = 10
      KMAXE = 2
      TKEJ = 10.0
      TKEX =  10.0
      IOFF = 1
      IMAP =  8
      JMAP =  5
C
C      THE FOLLOWING STATEMENTS DEFINE THE GEOMETRY FOR AN
C     ASSUMMED BODY SHAPE TO BE USED BY THE RAXJET CODE.
C
      IXY = 8
      XBT = 60.
      XM = 70.
      RJET = 7.09
      XSTRT = 60.
      XEND  = 120.
      ROB  = 30.
      DO I = 1, IXY
         XO(I) = XP(I)
         YO(I) = YP(I)
      END DO
      WRITE(11,100)
 100  FORMAT(' SUBROUTINE RAXSUB COMPLETED')
C
C********* THE FOLLOWING ARE TEMPORARY PARAMETERS TO ********
C********** MINIMIZE PAXJET RUN TIME                        ********
C
C     IMAX = 21
C     JMAX = 7
C     MIT = 3
C     NMAXJ = 21
C     XJENT = 5
C     IMAP = 4
C     JMAP = 3
C
      RETURN
      END
```

```
      SUBROUTINE PANEL
C
C
C         THIS PROGRAM (1) COMPUTES THE COORDINATES OF A NETWORK OF PANEL
C                          TO MODEL THE INVISCID PLUME SHAPE
C
C
C
C                      (2) COMPUTES THE INFLOW VELOCITIES AT EACH PANEL
C                           CENTER
C
C
      COMMON /HOWELL/ RSQ(30),RSQDUM,IGH,REFL
      COMMON /PEXTC/  IEXT,XEXT(30),PEXT(30),YPLU(30),PINT,XDK,UJET(30)
      COMMON /RAXOUT/ NPTS,XSTA(30),YRATIO(30),VN(29)
C
      DIMENSION VROU(29),VNOU(29),REFF(30)
C
C          XEXT  = X-STATIONS FOR YPLU, REFF, AND RSQ
C          YPLU  = RADIUS OF INVISCID PLUME
C          RSQ   = RADIUS SQUARED FOR EFF. PLUME (INCLUDES EFFECTS OF BOD
C          IGH   = NO. OF XEXT, YPLU, REFF, AND RSQ (IGH(1) CORRESPONDS A
C
C
C     MODIFY VALUES TO REFLECT INPUT UNITS
C
      DO 10  I = 1,IGH
      RSQ(I) = RSQ(I) / REFL**2
 10   CONTINUE
C
C
C
C****************** COMPUTE NORMAL VELOCITY **************************
C***************** AT INVISCID PLUME BOUNDARY ***********************
C
C
      IGHM1=IGH-1
      DO 20 I= 1,IGHM1
      XXX1 = RSQ(I) - YPLU(I) **2
      XXX2 = RSQ(I+1) - YPLU(I+1) **2
      XXX3 = (YPLU(I) + YPLU(I+1))/2
C     VROU = RADIAL VELOCITY
      VROU(I) = (XXX2 -XXX1)/XXX3
      VROU(I) = VROU(I)/ (XEXT(I+1) - XEXT(I))
C
C     THETA = SLOPE OF PANEL
      DY = YPLU(I+1) - YPLU(I)
      DX = XEXT(I+1) - XEXT(I)
      THETA = ATAN(DY/DX)
C
C     VNOU = VELOCITY NORMAL TO PANEL
      VNOU(I) = VROU(I) * COS(THETA)
 20   CONTINUE
C
C     REFF = EFFECTIVE BODY RADIUS
      DO 25 I=1,IGH
      REFF(I)=SQRT(ABS(RSQ(I)))
```

```
      IF(RSQ(1).LT.0.)  REFF(I)=-REFF(I)
 25   CONTINUE
C
C
C********** INSERT COMPUTED PARAMETERS INTO COMMON /RAXOUT/ ************
C
      NPTS = IGH
      DO 40   I = 1,30
         XSTA(I) = (XEXT(I)-XEXT(1))/ (2. * YPLU(1))
         YRATIO(I) = YPLU(I) /YPLU(1)
         IF(I.EQ.NPTS) GO TO 40
         VN(I) = VNOU(I)
 40   CONTINUE
C
C      PRINT RESULTS
C
      WRITE(6,500)
 500  FORMAT(1H1,/,6X,4HGRID,6X,2HX-,7X,6HRADIUS,5X,6HRADIUS,5X,
     1      8HVELOCITY,4X,8HVELOCITY)
      WRITE(6,501)
 501  FORMAT(5X,6HNUMBER,3X,7HSTATION,3X,8HINVISCID,3X,9HEFF-PLUME,
     1        5X,6HINFLOW,6X,6HRADIAL,/)
C
C
      DO 30  I = 1,IGH
      WRITE(6,510) I,XEXT(I),YPLU(I),REFF(I)
 510  FORMAT(7X,I3,3F11.3)
C
      IF(I.EQ.IGH) GO TO 30
      WRITE(6,520) VNOU(I), VROU(I)
 520  FORMAT(43X,2F11.3)
 30   CONTINUE
      END
```

APPENDIX C

USER'S GUIDE TO VISCOUS SIMULATION MODULE.

C.1 User Instructions For Interactive Vicsous Simulation Module

The user will be prompted with the following questions when using the viscous module described in Section 6.

(1) "Do you wish to start a new problem (Enter 1 for yes) and set up a new data base of flow properties?"
"or"
"No, I do not want to start a new problem (Enter 0 for No); I want to continue with previous data base." "Enter 0 or 1"

The user is given the option to continue with a data base already created from an PAN AIR problem or start by creating a new base. If "0" is entered control is shifted to question (4).

(2) "First Delete Previous Data Base Generated By Execution Of This Module." The user will first be asked to delete the previous data base (NETWORKS). If for some reason these are not to be deleted, the user must transfer them or rename them before entering the viscous module. Therefore, the answer to this question must be "YES" if the user wants to continue execution of the module. Actually the user will be asked to delete three files since a RIM data base consists of three files.

(3) "Enter Name of TAPE 10 Input File". The user must supply the name of the file containing the calculated flow properties which was transferred to the VAX system. This is the plot file that has been generated by PAN AIR.

(4) "Enter the Number of Networks on the Upper Surface That You Want To Combine For a Viscous Analysis (Enter 1 if the upper surface is composed of only 1 network)'. The user must enter an integer to specify the number of PAN

AIR networks that compose the upper surface. An index of the networks and their ID numbers will be displayed on the terminal to assist the user in answering this and the following questions.

(5) "Enter the Number of Networks on the Lower Surface That You Want To Combine For A Viscous Analysis". The same question as (3) must be answered for the lower surface.

(6) "Enter the Identification Numbers of the Upper Surface Networks". The user must supply the ID numbers of the upper surface networks. An index of these ID numbers and network titles will be displayed on the terminal.

(7) "Enter the Identification Numbers of the Lower Surface Networks". The same question as (5) must be answered for the lower surface.

(8) "Input the Desired Tolerance Dimension for Determining Station Cuts Along Surface". As described in Section 6, the panel center control points at any station on the surface may vary somewhat in the Y coordinate due to the paneling scheme. This tolerance tells the program what variance will be allowed in determining chordwise stations on the surface. The dimensions of the surface and paneling scheme should be considered in setting this tolerance. Caution: an unreasonably low tolerance will yield questionable results because the program will try to divide one chordwise station into several. In general, this tolerance should be at least 5 to 10 percent of the typical panel width.

(9) "Enter Free Stream Mach Number For Boundary Layer Analysis". User should input the Mach Number consistent with the PAN AIR problem.

(10) "Enter Unit Reynolds Numbers Per Million". Enter the Reynolds Number per unit length divided by 10^6. This Reynolds Number will be used in computation of the boundary layer characteristics.

(11) "First Delete the Previous Data Base Generated:". The user must delete the data base (ORDER) generated by the previous execution of this module. This data base is set up to reorder the panel boundary layer properties to be consistent with the PAN AIR network scheme.

(12) "Enter the Name of the File Containing the Original PAN AIR Problem Without Viscous Effects." Enter the initial PAN AIR problem which produced the inviscid flow properties for the boundary layer analysis. If more than one component is being analyzed (only one component can be analyzed each pass through this module), the new file name generated by the previous pass through this module should be entered.

(12) "Enter the File Name For the New PAN AIR Problem with Viscous Effects Simulated to Be Generated by This Program." Enter a file name for the new PAN AIR problem to be stored. This completed the execution of the Viscous Simulation Module. The new PAN AIR problem is ready to be run.

(13) "Do You Want To Do a Viscous Analysis For Another Surface?" The user may choose to analyze another surface in addition to the one just completed. Be sure to answer (12) with the file name containing the corrections for the previously analyzed surface.

The programs run during execution of the Viscous Simulation Module along with the files or data bases generated by each are shown in Table C-1.

C.2 VAX - Fortran Codes for Viscous Simulation Module

Enclosed within this section of the Appendix are copies of the codes of each routine developed specifically for this module. The program developed at NASA Ames to read the TAPE 10 output (EDITTEN), the RIM data base management program, and the Whitfield code were not developed specifically for this module and are not included. The other programs listed in Table C-1 are included in the following sections.

ORIGINAL PAGE IS
OF POOR QUALITY

Table C-1 PROGRAMS AND FILES GENERATED BY VISCOUS SIMULATION MODULE

Program	Files Generated	Purpose
EDITTEN	(1) PATEMP.COM	Command file to load RIM data base
	(2) RIMINFO.TMP	Information from PAN AIR run
	(3) INDEX.TMP	Index and ID no. of networks
RIM	(1) NETWKS 1.DAT	RIM data base
	(2) NETWKS 2.DAT	RIM data base
	(3) NETWKS 3.DAT	RIM data base
	(4) LABEL.TMP	Comments from RIM execution
SELECT	(1) RUNRIM.COM	Command file to open RIM data base and select flow properties
RIM	(1) WINGU.DAT	Flow properties for upper surface
	(2) WINGL. DAT	Flow properties for lower surface
	(3) LABEL.TMP	Comments from RIM execution
WHITIN	(1) UPPER.DAT	Whitfield input problem
	(2) LOWER.DAT	Whitfield input problem
	(3) STAG. DAT	Stagnation points
WHITFIELD	(1) UPROUT.DAT	Upper surface B.L. predictions
	(2) LOWOUT.DAT	Lower surface B.L. predictions
WHITOUT	(1) REORDER.COM	Command file to load RIM data bas
RIM	(1) ORDER 1.DAT	RIM data base
	(2) ORDER 2.DAT	RIM data base
	(3) ORDER 3.DAT	RIM data base
	(4) LABEL.TMP	RIM information
SORT	(1) RIMSORT.COM	Command file to open RIM data bas and reorder parameters
RIM	(1) LINK.DAT	Reorder network parameters
LINK	NEW PAN AIR PROBLEM	

C.2.1 Code Developed For Program Select

The following pages contain the FORTRAN code developed for the program SELECT.

ORIGINAL PAGE IS
OF POOR QUALITY

```
      PROGRAM SELECT
C  THIS PROGRAM ASKS THE USER TO IDENTIFY THE UPPER AND LOWER
C  NETWORKS THAT ARE GOING TO BE USED IN THE VISCOUS ANALYSIS.
C  OFTEN THE WING IS DIVIDED INTO SEVERAL NETWORKS THUS IT IS
C  NECESSARY TO COMBINE THESE NETWORKS TO IDENTIFY THE
C    PAN AIR PREDICTED FLOW PARAMETERS FROM LEADING
C   TO TRAILING EDGE FOR THE ENTIRE SPAN OF THE WING.
C  THESE FLOW PARAMETERS HAVE BEEN STORED IN A RIM MANAGED
C   DATA BASE . BY IDENTIFYING THE UPPER AND LOWER
C   NETWORKS REQUIRED FOR THE VISCOUS CALCULATIONS
C  A COMMAND FILE TO RUN RIM AND SELECT THE REQUIRED
C   DATA MAY BE WRITTEN .
C  THIS PROGRAM WRITES THAT FILE (RUNRIM.COM).
      DIMENSION NETU(20),NETL(20)
      WRITE(6,*)'  ENTER THE NUMBER OF NETWORKS ON THE UPPER'
      WRITE(6,*)'  SURFACE THAT YOU WANT TO COMBINE FOR A'
      WRITE(6,*)'  VISCOUS ANALYSIS  ( ENTER 1 IF THE UPPER'
      WRITE(6,*)' SURFACE IS COMPOSED OF ONLY 1 NETWORK)   '
      READ(5,10) NNETU
10    FORMAT(I2)
      WRITE(6,*)'     '
      WRITE(6,*)'  ENTER THE NUMBER OF NETWORKS ON THE LOWER'
      WRITE(6,*)'  SURFACE THAT YOU WANT TO COMBINE FOR A'
      WRITE(6,*)'  VISCOUS ANALYSIS  (INTERGER FORMAT). ENTER 1 IF  '
       WRITE(6,*)'  THE LOWER SURFACE IS COMPOSED OF ONLY 1 NETWORK'
      READ(5,10) NNETL
      OPEN(UNIT=2,FILE='RUNRIM.COM',STATUS='NEW')
      WRITE(2,15)
15    FORMAT(
     2' $RUN [BHATELEY.PANAIR]RIM23'/
     3' *(SET SEMI=NULL)'/
     4' *(SET DOLLAR=NULL)'/
     5' NOECHO'/
     6' OPEN NETWKS'/
     8' LINES 2000'/
     9' OUTPUT WINGU')
C WRITE OUT UPPER SURFACE COMMANDS
      WRITE(2,16)
16    FORMAT(
     2' SELECT NN,X,Y,Z,MLISEN,RN,CN FROM NETWKS  +'/
     3' SORTED BY Y X  + ')
      WRITE(6,*)'  NOW ENTER THE IDENTIFICATION NUMBERS OF THE'
      WRITE(6,*)'  UPPER SURFACE NETWORKS REQUIRED FOR THE VISCOUS  '
      WRITE(6,*)'  ANALYSIS. ENTER THEM IN INTERGER FORM . ON ONE LINE'
      WRITE(6,*)'  SEPERATED BY COMMAS AS IN THE FOLLOWING EXAMPLE   '
      WRITE(6,*)'  1,2,3,4 ETC.   '
      READ(5,18) (NETU(I) , I=1,NNETU)
18    FORMAT(20I3)
      IF(NNETU.EQ.1) THEN
      WRITE(2,19) NETU(1)
19    FORMAT(
     2' WHERE NN EQ ',I3,'   + '/
     3' AND Y EQA Y ')
      GO TO 27
      END IF
      DO 25 I=1,(NNETU-1)
      IF(I.GT.1) GO TO 21
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      WRITE(2,20) NETU(1)
20    FORMAT(
     2' WHERE NN EQ ',I3,' OR  +')
      GO TO 25
21    CONTINUE
      WRITE(2,22) NETU(1)
22    FORMAT(
     2' NN EQ ',I3,' OR  +')
25    CONTINUE
      WRITE(2,26) NETU(NNETU)
26    FORMAT(
     2'  NN EQ ',I3,'  + '/
     3' AND Y EQA Y ')
27    CONTINUE
C  NOW WRITE OUT LOWER SURFACE:::::::
      WRITE(2,30)
30    FORMAT(
     2' OUTPUT WINGL '/
     3' SELECT NN,X,Y,Z,MLISEN,RN,CN FROM NETWKS  +'/
     4' SORTED BY Y X=D  +')
      WRITE(6,*)'  NOW ENTER THE IDENTIFICATION NUMBERS OF'
      WRITE(6,*)'  THE LOWER SURFACE NETWORKS. ENTER THEM'
      WRITE(6,*)'  IN INTERGER FORM , ON ONE LINE , SEPERATED'
      WRITE(6,*)'  BY COMMAS AS IN THE FOLLOWING EXAMPLE 5,6,7,8'
      READ(5,32) (NETL(I),I=1,NNETL)
32    FORMAT(20I3)
      IF(NNETL.EQ.1) THEN
      WRITE(2,33) NETL(1)
33    FORMAT(
     2' WHERE NN EQ ',I3,'  + '/
     3' AND Y EQA Y '/
     4' EXIT ')
      GO TO 44
      END IF
      DO 40 J=1,(NNETL-1)
      IF(J.GT.1) GO TO 35
      WRITE(2,34) NETL(J)
34    FORMAT(
     2' WHERE NN EQ ',I3,'  OR  + ')
      GO TO 40
35    CONTINUE
      WRITE(2,36) NETL(J)
36    FORMAT(
     2'  NN EQ ',I3,'  OR  + ')
40    CONTINUE
      WRITE(2,42) NETL(NNETL)
42    FORMAT(
     2'  NN EQ ',I3,'  + '/
     3' AND Y EQA Y '/
     4' EXIT')
44    CONTINUE
      STOP
      END
```

C.2.2 Code Developed For Program WHITIN

The following pages contain the FORTRAN code developed for the program WHITIN.

```
      PROGRAM WHITIN
C
C
C    THIS PROGRAM COMBINES
C    TWO FILES WHICH CONTAIN PANEL COORDINATE DATA
C    AND LOCAL MACH NUMBER FOR THE SURFACE UPPER
C    AND LOWER PANELS . THESE FILES WERE GENERATED
C    USING THE RIM DATA MANAGEMENT SYSTEM .
C    THE STAGNATION POINT IS LOCATED AND TWO OUTPUT FILES
C    ARE GENERATED ( UPPER.DAT AND LOWER.DAT ) .THE UPPER
C    FILE CONTAINS THE POINTS FORWARD OF THE STAG. POINT
C    ON THE LOWER SURFACE AND THE UPPER SURFACE POINTS
C    TO THE TRAILING EDGE . THE LOWER FILE CONTAINS THOSE
C    POINTS FROM THE STAG. POINT TO THE TRAILING EDGE .
C    THESE FILES ARE IN A FORMAT SUITABLE FOR INPUT INTO
C    THE WHITFIELD BOUNDARY LAYER PROGRAM .
C
C
      IMPLICIT REAL(M)
      DIMENSION X(20,100),Y(20,100),Z(20,100),JMAX(20),JMAXL(20)
      DIMENSION ML(20,100),INDST(20),NN(20,100),IRN(20,100),ICN(20,100)
      DIMENSION SU(20,100),SL(20,100),NPTSU(20),NPTL(20),ISYM(20)
      DIMENSION XX(200),YY(200),ZZ(200),NNN(200),MML(200),IIRN(200),
     2 IICN(200)
      CHARACTER*8 FU(20),FL(20)
      CHARACTER*13 UPPER,LOWER
      CHARACTER FNAME*13,LINE*80,NO*2
      DATA X,Y,Z,ML / 2000*0.0,2000*0.0,2000*0.0,2000*0.0/
      DATA IRN,ICN / 2000*0.0,2000*0.0/
      DATA XX,YY,ZZ / 200*0.0,200*0.0,200*0.0 /
      DATA MML,IIRN,IICN / 200*0.0,200*0.0,200*0.0 /
      WRITE(6,*)'  INPUT THE DESIRED TOLERANCE DIMENSION FOR'
      WRITE(6,*)'  DETERMINING STATION CUTS ALONG SURFACE '
      WRITE(6,*)'          '
      WRITE(6,*)'        TOLERANCE = ?   '
      READ(5,*) TOL
C     WRITE(6,*)'  WHAT IS THE NAME OF THE FILE CONTAINING'
C     WRITE(6,*)'  THE UPPER SURFACE DATA ??? '
C     READ(5,11) UPPER
11    FORMAT(A13)
C     WRITE(6,*)'   WHAT IS THE NAME OF THE FILE CONTAINING'
C     WRITE(6,*)'   THE LOWER SURFACE DATA ???'
C     READ(5,11) LOWER
      OPEN(UNIT=1,FILE='WINGU.DAT',STATUS='OLD')
      OPEN(UNIT=2,FILE='WINGL.DAT',STATUS='OLD')
C READ THE TITLE CARDS
      DO 4 K=1,3
      READ(1,3)LINE
      READ(2,3)LINE
3     FORMAT(A80)
4     CONTINUE
      CHECK=0.0
      I=1
      J=1
5     CONTINUE
      READ(2,10,END=30) NN(I,J),X(I,J),Y(I,J),Z(I,J),ML(I,J),IRN(I,J),
     1  ICN(I,J)
```

```
10    FORMAT(I10,4F10.5,2I10)
      IF(J.EQ.1) GO TO 20
      DY=ABS(Y(I,J)-Y(I,J-1))
      IF(DY.LE.TOL) THEN
      IF(DY.GT.0.0) CHECK=1.0
      IF(X(I,J).LT.X(I,J-1)) CHECK=1.0
      GO TO 20
      END IF
      JMAXL(I)=J-1
      IF (CHECK.EQ.1.0) THEN
      CALL CKLOWER(I,J,JMAXL,NN,X,Y,Z,ML,IRN,ICN,CHECK)
      CHECK=0.0
      END IF
      I=I+1
      J=1
      BACKSPACE 2
      GO TO 5
20    CONTINUE
      J=J+1
      GO TO 5
30    CONTINUE
      JMAXL(I)=J-1
      IMAXU=I
      IF(CHECK.EQ.1.0)THEN
      CALL CKLOWER(I,J,JMAXL,NN,X,Y,Z,ML,IRN,ICN,CHECK)
      CHECK=0.0
      END IF
      I=1
      J=JMAXL(1)+1
50    CONTINUE
      READ(1,10,END=100) NN(I,J),X(I,J),Y(I,J),Z(I,J),ML(I,J),
     1   IRN(I,J),ICN(I,J)
      IF(J.EQ.JMAXL(I)+1) GO TO 60
      DY=ABS(Y(I,J)-Y(I,J-1))
      IF(DY.LE.TOL) THEN
      IF(DY.GT.0.0) CHECK=1.0
      IF(X(I,J).LT.X(I,J-1)) CHECK=1.0
      GO TO 60
      END IF
      JMAX(I)=J-1
      IF (CHECK.EQ.1.0) THEN
      CALL CKUPPER(I,J,JMAXL,JMAX,NN,X,Y,Z,ML,IRN,ICN,CHECK)
      CHECK=0.0
      END IF
      I=I+1
      BACKSPACE 1
      J=JMAXL(I)+1
      GO TO 50
60    CONTINUE
      J=J+1
      GO TO 50
100   CONTINUE
      JMAX(I)=J-1
      IF(CHECK.EQ.1.0) THEN
      CALL CKUPPER(I,J,JMAXL,JMAX,NN,X,Y,Z,ML,IRN,ICN,CHECK)
      CHECK=0.0
      END IF
```

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

```
      IMAX=1
      IF(IMAXU.EQ.IMAX) GO TO 110
      GO TO 130
110   CONTINUE
      WRITE(6,*)' POTENTIAL ERROR EXISTS IN DATA  '
      WRITE(6,*)' FROM INPUT FILES ---PLEASE CHECK FILES '
      WRITE(6,*)'  IMAXU=1 , IE. ONLY 1 WING STATION EXISTS ! '
      WRITE(6,*)'         '
130   CONTINUE
C
C   CHECK FOR MACH NO. = 0.0
C
      I=1
      J=1
      DO 134 I=1,IMAXU
      DO 132 J=1,JMAX(I)
      IF(ML(I,J).EQ.0.0) THEN
      ML(I,J)=(ML(I,J-1)+ML(I,J+1))/2.0
      WRITE(6,*) '          '
      WRITE(6,*) '  A LOCAL MACH NO. EQUAL TO 0 WAS FOUND. '
      WRITE(6,*) ' THIS MACH NO. WAS SET = TO 0 BY PAN AIR WHEN '
      WRITE(6,*) ' VACUUM PRESSURE WAS REACHED .'
      WRITE(6,*) ' AN INTERPOLATED VALUE OF MACH NO. WILL'
      WRITE(6,*) ' BE USED FOR BOUNDARY LAYER ANALYSIS .'
      WRITE(6,*) '         '
      END IF
132   CONTINUE
134   CONTINUE
      I=1
      J=1
      DO 150 I=1,IMAXU
      JM=JMAX(I)
      KM=JMAXL(I)
      DO 140 J=1,JM
C     WRITE(3,135) X(I,J),Y(I,J),Z(I,J),ML(I,J)
135   FORMAT(4F10.5)
140   CONTINUE
150   CONTINUE
      GO TO 800
500   WRITE(6,*)' ERROR EXISTS IN FILE CONTAINING '
      WRITE(6,*)' DATA FOR LOWER SURFACE '
      GO TO 800
501   WRITE(6,*)' ERROR EXISTS IN FILE CONTAINING '
      WRITE(6,*)' DATA FOR UPPER SURFACE '
800   CONTINUE
      DO 162 I=1,IMAXU
      ISYM(I)=0
      CHORD=X(I,1)-X(I,JMAXL(I))
      CH30=.30*CHORD
      INDST(I)=JMAXL(I)
      MLMIN=ML(I,KM)
      JM=JMAX(I)
      DO 160 J=1,JM
      IF(J.EQ.1) GO TO 160
      IF((X(I,J)-X(I,JMAXL(I))).GE.CH30) GO TO 160
      IF(ML(I,J).GT.MLMIN) GO TO 160
      MLMIN=ML(I,J)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      INDST(I)=J
160   CONTINUE
C
C   CHECK FOR SYMETRICAL DISTRIBUTION
C
      TOLER=0.02
      CKTOL=ABS(ML(I,KM)-ML(I,KM+1))
      CKML=ABS(ML(I,KM)-MLMIN)
      IF(TOLER.GT.CKTOL.AND.TOLER.GT.CKML) ISYM(I)=1
      IF(ISYM(I).EQ.1) THEN
      INDST(I)=JMAXL(I)+1
      END IF
162   CONTINUE
      OPEN(UNIT=7,FILE='UPPER.DAT',STATUS='NEW')
      OPEN(UNIT=8,FILE='LOWER.DAT',STATUS='NEW')
      WRITE(7,200) IMAXU
      WRITE(8,200) IMAXU
200   FORMAT(I5)
      JJJ=0
      DO 250 I=1,IMAXU
204   CONTINUE
C
C   GENERATE UPPER FILE
C
      IF(I.GT.1) GO TO 201
      WRITE(6,*)' ENTER FREE STREAM MACH NUMBER FOR BOUNDARY'
      WRITE(6,*)' LAYER ANALYSIS '
      WRITE(6,*)'          MACH = ???   '
      READ(5,*)MACH
      WRITE(6,*)'ENTER UNIT REYNOLDS NUMBER PER MILLION'
      WRITE(6,*)'     REN = ????? (PER 10E6) '
      READ(5,*)REN
      THETA=0.0
      SHAPE=0.0
201   CONTINUE
      WRITE(7,202) MACH,REN,THETA,SHAPE
      WRITE(8,202) MACH,REN,THETA,SHAPE
202   FORMAT(4F10.5)
      NPTSU(I)=JMAX(I)-INDST(I)
      S=0.0
      WRITE(7,205) NPTSU(I)
205   FORMAT(I3)
      DO 210 J=INDST(I),JMAX(I)
      IF(J.EQ INDST(I)) GO TO 210
      KK=J-1
      S=S+SQRT((X(I,J)-X(I,KK))**2+(Y(I,J)-Y(I,KK))**2+
     1   (Z(I,J)-Z(I,KK))**2)
      RR=1.0
      WRITE(7,206) S,ML(I,J),RR,NN(I,J),X(I,J),Y(I,J),Z(I,J),
     1  IRN(I,J),ICN(I,J)
206   FORMAT(3F10.5,I5,3F10.5,2I5)
210   CONTINUE
C
C   GENERATE LOWER FILE
C
      IF(ISYM(I).EQ.1) THEN
      INDST(I)=JMAXL(I)
```

```
      END IF
      NPTL(I)=INDST(I)-1
      WRITE(8,205) NPTL(I)
      S=0.0
      DO 220 J=INDST(I),1,-1
      IF(J.EQ.INDST(I)) GO TO 220
      KK=J+1
      S=S+SQRT((X(I,J)-X(I,KK))**2+(Y(I,J)-Y(I,KK))**2+
     1    (Z(I,J)-Z(I,KK))**2)
      RR=1.0
      IF(RR.LE.0.0)THEN
      IF(JJJ.EQ.1) GO TO 216
      WRITE(6,*)'          '
      WRITE(6,*)'         '
      WRITE(6,*)'     WARNING : THE WHITFIELD BOUNDARY LAYER '
      WRITE(6,*)'                    PROGRAM WILL NOT WORK WITH '
      WRITE(6,*)'                      NEGATIVE VALUES ( OR ZERO ) '
      WRITE(6,*)'                    OF SURFACE ORDINATES .'
      WRITE(6,*)'  YOUR SURFACE CONTAINS SUCH VALUES THEREFORE '
      WRITE(6,*)'  THE SIGNS ARE BEING CHANGED FOR THE LOWER '
      WRITE(6,*)'  BOUNDARY LAYER ANALYSIS . THERE WILL NOT'
      WRITE(6,*)'  BE A PROBLEM IF ALL LOWER SURFACE ORDINATES'
      WRITE(6,*)'  ARE OF THE SAME SIGN.'
      JJJ=1
216   CONTINUE
      RR=-Z(I,J)
      END IF
      WRITE(8,206) S,ML(I,J),RR,NN(I,J),X(I,J),Y(I,J),Z(I,J),
     1  IRN(I,J),ICN(I,J)
220   CONTINUE
250   CONTINUE
      CLOSE(UNIT=7)
      CLOSE(UNIT=8)
C
C     GENERATE A FILE CONTAINING STAGNATION POINTS
C
      OPEN(UNIT=9,FILE='STAG.DAT',STATUS='NEW')
      DO 300 II=1,IMAXU
      SS=0.0
      SLOPE=0.0
      RR=0.0
      KK=INDST(II)
      WRITE(9,213) SS,ML(II,KK),NN(II,KK),X(II,KK),
     1   Y(II,KK),Z(II,KK),IRN(II,KK),ICN(II,KK),SLOPE
213   FORMAT(2E12.5,I5,3E12.5,2I5,1E12.5)
      IF(ISYM(II).EQ.1)THEN
      KK=JMAXL(II)+1
      WRITE(9,213) SS,ML(II,KK),NN(II,KK),X(II,KK),
     2  Y(II,KK),Z(II,KK),IRN(II,KK),ICN(II,KK),SLOPE
      END IF
300   CONTINUE
      CLOSE(UNIT=9)
      WRITE(6,*)'  THREE FILES HAVE NOW BEEN GENERATED :'
      WRITE(6,*)'         1 : UPPER.DAT '
      WRITE(6,*)'         2 : LOWER.DAT '
      WRITE(6,*)'         3 : STAG.DAT '
      WRITE(6,*)'  UPPER.DAT AND LOWER.DAT ARE IN THE FORMAT '
```

```
      WRITE(6,*)'  REQUIRED FOR INPUT INTO THE WHITFIELD '
      WRITE(6,*)'  BOUNDARY LAYER PROGRAM '
      WRITE(6,*)'  STAG.DAT CONTAINS THE PANEL POINTS DEFINED'
      WRITE(6,*)'  AS STAGNATION POINTS ; THIS FILE WILL BE '
      WRITE(6,*)'  USED LATER TO GENERATE THE BOUNDARY CONDITIONS'
      WRITE(6,*)'  AFTER THE OTHER TWO FILES HAVE BEEN ANALYZED'
      WRITE(6,*)'  IN THE WHITFIELD PROGRAM.'
      STOP
      END
      SUBROUTINE CKLOWER(I,J,JMAXL,NN,X,Y,Z,ML,IRN,ICN,CHECK)
C
C  ALL DATA SELECTED FROM RIM DATA BASE WAS SORTED ON THE BASIS OF EQUAL
C  VALUES OF Y . SOMETIMES A SMALL VARIANCE IN Y WILL CAUSE THE ORDER TO
C  ERROR . WHEN THIS CONDITION IS DETECTED THIS SUBROUTINE IS EXECUTED T
C
C
      DIMENSION X(20,100),Y(20,100),Z(20,100),JMAXL(20)
      DIMENSION ML(20,100),NN(20,100),IRN(20,100),ICN(20,100)
      DIMENSION XX(200),YY(200),ZZ(200),NNN(200),MML(200),IIRN(200),
     2 IICN(200)
C  THIS IS REORDERING A LOWER SURFACE SO THE X VALUES MUST BE IN
C   DESCENDING ORDER .
C
      DO 30 KK=1,JMAXL(I)
      IF(KK.GT.1) GO TO 15
C
C FIRST FIND LARGEST VALUE OF X .
C
      JTEST=1
      XTEST=X(I,1)
      DO 10 JJ=1,JMAXL(I)
      IF(X(I,JJ).GT.XTEST) THEN
      XTEST=X(I,JJ)
      JTEST=JJ
      END IF
10    CONTINUE
      XX(1)=X(I,JTEST)
      YY(1)=Y(I,JTEST)
      ZZ(1)=Z(I,JTEST)
      NNN(1)=NN(I,JTEST)
      MML(1)=ML(I,JTEST)
      IIRN(1)=IRN(I,JTEST)
      IICN(1)=ICN(I,JTEST)
      GO TO 30
15    CONTINUE
C
C  NOW FIND NEXT LARGEST VALUE OF X
C
      JJTEST=1
      XXTEST=X(I,1)
      DO 20 JJ=1,JMAXL(I)
      IF(XXTEST.GE.XX(KK-1)) THEN
      XXTEST=X(I,JJ+1)
      JJTEST=JJ+1
      GO TO 20
      END IF
      IF(X(I,JJ).LT.XX(KK-1).AND.X(I,JJ).GT.XXTEST) THEN
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      XXTEST=X(I,JJ)
      JJTEST=JJ
      END IF
20    CONTINUE
      XX(KK)=X(I,JJTEST)
      YY(KK)=Y(I,JJTEST)
      ZZ(KK)=Z(I,JJTEST)
      NNN(KK)=NN(I,JJTEST)
      MML(KK)=ML(I,JJTEST)
      IIRN(KK)=IRN(I,JJTEST)
      IICN(KK)=ICN(I,JJTEST)
30    CONTINUE
C
C  CORRECT ORDER HAS NOW BEEN DETERMINED SO REORDER THE VALUES.
C
C     WRITE(6,*)'    I = ',I
      DO 40 KKK=1,JMAXL(I)
      X(I,KKK)=XX(KKK)
      Y(I,KKK)=YY(KKK)
      Z(I,KKK)=ZZ(KKK)
C     WRITE(6,*)'  X = ',X(I,KKK),' Y = ',Y(I,KKK),'     Z = ',Z(I,KKK)
      NN(I,KKK)=NNN(KKK)
      ML(I,KKK)=MML(KKK)
      IRN(I,KKK)=IIRN(KKK)
      ICN(I,KKK)=IICN(KKK)
40    CONTINUE
      CHECK=0.0
      RETURN
      END
      SUBROUTINE CKUPPER(I,J,JMAXL,JMAX,NN,X,Y,Z,ML,IRN,ICN,CHECK)
C
C  ALL DATA SELECTED FROM RIM DATA BASE WAS SORTED ON THE BASIS OF EQUAL
C  VALUES OF Y . SOMETIMES A SMALL VARIANCE IN Y WILL CAUSE THE ORDER TO
C  ERROR . WHEN THIS CONDITION IS DETECTED THIS SUBROUTINE IS EXECUTED T
C
C
      DIMENSION X(20,100),Y(20,100),Z(20,100),JMAXL(20),JMAX(20)
      DIMENSION ML(20,100),NN(20,100),IRN(20,100),ICN(20,100)
      DIMENSION XX(200),YY(200),ZZ(200),NNN(200),MML(200),IIRN(200),
     2 IICN(200)
C  THIS IS REORDERING A UPPER SURFACE SO THE X VALUES MUST BE IN
C   ASCENDING ORDER .
C
      I1=JMAXL(I)+1
      I2=JMAX(I)
C     WRITE(6,*)'  I = ',I,'    I1 = ',I1,'    I2 = ',I2
      DO 30 KK=I1,I2
      IF(KK.GT.I1) GO TO 15
C
C FIRST FIND SMALLEST VALUE OF X .
C
      JTEST=I1
      XTEST=X(I,I1)
      DO 10 JJ=I1,I2
      IF(X(I,JJ).LT.XTEST) THEN
      XTEST=X(I,JJ)
      JTEST=JJ
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      END IF
10    CONTINUE
      XX(I1)=X(I,JTEST)
      YY(I1)=Y(I,JTEST)
      ZZ(I1)=Z(I,JTEST)
      NNN(I1)=NN(I,JTEST)
      MML(I1)=ML(I,JTEST)
      IIRN(I1)=IRN(I,JTEST)
      IICN(I1)=ICN(I,JTEST)
      GO TO 30
15    CONTINUE
C
C  NOW FIND NEXT SMALLEST VALUE OF X
C
      JJTEST=I1
      XXTEST=X(I,I1)
      DO 20 JJ=I1,I2
      IF(XXTEST.LE.XX(KK-1)) THEN
      XXTEST=X(I,JJ+1)
      JJTEST=JJ+1
      GO TO 20
      END IF
      IF(X(I,JJ).GT.XX(KK-1).AND.X(I,JJ).LT.XXTEST) THEN
      XXTEST=X(I,JJ)
      JJTEST=JJ
      END IF
20    CONTINUE
      XX(KK)=X(I,JJTEST)
      YY(KK)=Y(I,JJTEST)
      ZZ(KK)=Z(I,JJTEST)
      NNN(KK)=NN(I,JJTEST)
      MML(KK)=ML(I,JJTEST)
      IIRN(KK)=IRN(I,JJTEST)
      IICN(KK)=ICN(I,JJTEST)
30    CONTINUE
C
C  CORRECT ORDER HAS NOW BEEN DETERMINED SO REORDER THE VALUES.
C
      DO 40 KKK=I1,I2
      X(I,KKK)=XX(KKK)
      Y(I,KKK)=YY(KKK)
      Z(I,KKK)=ZZ(KKK)
C     WRITE(6,*)'    X = ',X(I,KKK),' Y = ',Y(I,KKK),' Z = ',Z(I,KKK)
      NN(I,KKK)=NNN(KKK)
      ML(I,KKK)=MML(KKK)
      IRN(I,KKK)=IIRN(KKK)
      ICN(I,KKK)=IICN(KKK)
40    CONTINUE
      CHECK=0.0
      RETURN
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

C.2.3. Code Developed For Program WHITOUT

The following pages contain the FORTRAN code developed for the program WHITOUT.

```
C
C   THIS PROGRAM TAKES THREE FILES ( UPPER.OUT,LOWER.OUT,AND
C   STAG.DAT AND LOADS A COMMAND FILE TO GENERATE A RIM DATA
C   BASE.
C
      PROGRAM WHITOUT
      CHARACTER*5 DBNAME
      CHARACTER*13 FUPPER,FLOWER
      DIMENSION S(2000),DELSTAR(2000),NN(2000),SLOPE(2000)
     2 ,X(2000),Y(2000),Z(2000),IRN(2000),ICN(2000)
10    FORMAT(A13)
12    FORMAT(2E12.5,I5,3E12.5,2I5,1E12.5)
      DBNAME(1:5)='ORDER'
C     WRITE(6,*)' ENTER NAME OF UPPER SURFACE OUTPUT FILE'
C     WRITE(6,*)' FROM WHITFIELD PROGRAM (13 CHARACTERS MAX.)'
C     READ(5,10) FUPPER
C     WRITE(6,*)' ENTER NAME OF LOWER SURFACE OUTPUT FILE '
C     WRITE(6,*)' FROM WHITFIELD PROGRAM (13 CHARACTERS MAX.)'
C     READ(5,10) FLOWER
      OPEN(UNIT=1,FILE='STAG.DAT',STATUS='OLD')
      OPEN(UNIT=2,FILE='UPROUT.DAT',STATUS='OLD')
      OPEN(UNIT=3,FILE='LOWOUT.DAT',STATUS='OLD')
      JJ=0
      KKK=0
5     DO WHILE(KKK.EQ.0)
      JJ=JJ+1
      READ(1,12,IOSTAT=KKK,END=8) S(JJ),DELSTAR(JJ),NN(JJ),
     1  X(JJ),Y(JJ),Z(JJ),IRN(JJ),ICN(JJ),SLOPE(JJ)
      END DO
8     KKK=0
      JJ=JJ-1
15    DO WHILE(KKK.EQ.0)
      JJ=JJ+1
      READ(2,12,IOSTAT=KKK,END=18) S(JJ),DELSTAR(JJ),NN(JJ),
     1 X(JJ),Y(JJ),Z(JJ),IRN(JJ),ICN(JJ),SLOPE(JJ)
      END DO
18    KKK=0
      JJ=JJ-1
20    DO WHILE(KKK.EQ.0)
      JJ=JJ+1
      READ(3,12,IOSTAT=KKK,END=28) S(JJ),DELSTAR(JJ),NN(JJ),
     1  X(JJ),Y(JJ),Z(JJ),IRN(JJ),ICN(JJ),SLOPE(JJ)
      END DO
28    KKK=0
      JJ=JJ-1
C
C   NOW WRITE COMMAND FILE TO CREATE RIM DATA BASE
C
      OPEN(UNIT=4,FILE='REORDER.COM',STATUS='NEW')
      WRITE(4,30) DBNAME
30    FORMAT(
     2' $ RUN EBHATELEY.PANAIR]RIM23'/
     3' *(SET SEMI=NULL)'/
     4' *(SET DOLLAR=NULL)'/
     5' ECHO'/
     6' DEFINE ',A,/
     7' OWNER "NONE"'/
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
     8' ATTRIBUTES'/
     9' NN        INT')
      WRITE(4,31)
31    FORMAT(
     2' RN        INT'/
     3' CN        INT'/
     4' S         REAL'/
     5' DELSTAR   REAL'/
     6' X         REAL'/
     7' Y         REAL'/
     8' Z         REAL'/
     9' SLOPE     REAL')
      WRITE(4,32) DBNAME
32    FORMAT(' RELATIONS '/1X,A,1X,' WITH     +'/
     2' S         DELSTAR     NN        X       +',/
     3' Y       Z         RN            CN        SLOPE        ')
      WRITE(4,33)
33    FORMAT(' PASSWORDS '/' END'/' NOCHECK')
      WRITE(4,35) DBNAME
35    FORMAT(' LOAD ',A)
      II=0
36    DO WHILE(II.LT.JJ)
      II=II+1
      WRITE(4,37) S(II),DELSTAR(II),NN(II)
37    FORMAT(1X,2(F15.5,1X),I5,3X,'+')
      WRITE(4,38) X(II),Y(II),Z(II),IRN(II),ICN(II)
38    FORMAT(1X,3(F15.5,1X),2(I5,1X),'+')
      WRITE(4,39) SLOPE(II)
39    FORMAT(1F15.7)
      END DO
      WRITE(4,898)
898   FORMAT(' END'/' INPUT INPUT'/' EXIT')
      CLOSE(UNIT=4)
C     WRITE(6,891)
891   FORMAT(' THE FILE REORDER.COM HAS BEEN WRITTEN.'
     2 /' THIS IS A COMMAND FILE THAT CAN BE BEST EXECUTED'
     3 /' BY USING FORK .'/
     4'   JUST DO THE FOLLOWING : '/
     5'           1.TYPE IN FORK<CR> (YOU WILL GET A "COMMANDS:"
     6   PROMPT)'/
     7'           2.TYPE IN $ @REORDER<CR>'/
     8'  JUST BE SURE NOT TO LOG OFF BEFORE FORK TELLS YOU
     9 YOUR JOB IS FINISHED')
      STOP
      END
```

C.2.4 Code Developed For Program SORT

The following pages contain the FORTRAN code developed for the program SORT.

ORIGINAL PAGE IS
OF POOR QUALITY

```
      PROGRAM SORT
C
C
C     THIS PROGRAM WRITES A COMMAND FILE ( RIMSORT.COM ) WHICH OPENS
C     OPENS THE RIM DATA BASE NAMED ORDER AND USES RIM COMMANDS
C      TO SELECT AND SORT THE RESULTS FROM THE WHITFIELD PROGRAM
C     TO A FORMAT USEFUL TO GENERATE A NEW PAN AIR PROBLEM SIMULATING
C     VISCOUS EFFECTS.
C
C
      OPEN(UNIT=2,FILE='RIMSORT.COM',STATUS='NEW')
      WRITE(2,10)
10    FORMAT(
     2' $RUN [BHATELEY.PANAIR]RIM23'/
     3' *(SET SEMI=NULL)'/
     4' *(SET DOLLAR=NULL)'/
     5' NOECHO '/
     6' OPEN ORDER '/
     7' LINES 2000 '/
     8' OUTPUT LINK '/
     9' SELECT NN,X,Y,Z,S,DELSTAR,RN,CN,SLOPE FROM ORDER  + ')
      WRITE(2,15)
15    FORMAT(
     2'SORTED BY NN CN RN  '/
     3' EXIT  ')
      STOP
      END
FORTRAN STOP
$
```

C.2.5 Code Developed For Program LINK

The following pages contain the code developed for the program LINK.

ORIGINAL PAGE IS
OF POOR QUALITY

```
      PROGRAM LINK
C   THIS PROGRAM READS THE BL CHARACTERISTICS PREDICTED BY THE
C   WHITFIELD PROGRAM AND THE ORIGINAL INVISCID PAN AIR PROBLEM
C   THEN PRODUCES A NEW PAN AIR PROBLEM WHICH CONTAINS VISCOUS
C   CORRECTIONS IN THE FORM OF BOUNDARY CONDITIONS (OUTFLOW
C   VELOCITIES SPECIFIED AT CONTROL POINTS)
C
C     THE BOUNDARY LAYER CHARACTERISTICS ARE ASSUMED TO BE
C     STORED IN A FILE WHICH WAS GENERATED USING RIM COMMANDS
C     ON A RIM DATA BASE PRODUCED BY WHIT.COM FILE. THE
C     PROPER RIM COMMAND TO PRODUCE THIS DATA FILE IS :
C
C     SELECT NN,X,Y,Z,S,DELSTAR,RN,CN,SLOPE FROM DBNAME +
C     SORTED BY NN CN RN
C
C     WHERE DBNAME IS THE NAME OF THE RELATION CONTAINING
C     THE BL PARAMETERS.
C THE FILE MUST CONTAIN THE ABOVE SELECTED PARAMETERS AND
C MUST BE SORTED AS INDICATED.
C
      CHARACTER LINE*80,BC*4,PAFILE*20,RMFILE*13,NEWFILE*20
      DIMENSION NN(500),X(500),Y(500),Z(500),DELSTAR(500),IRN(500)
      DIMENSION ICN(500),S(500),SLP(500)
      DIMENSION XPA(20,20),YPA(20,20),ZPA(20,20)
      DIMENSION S1(20),S2(20),S3(20),S4(20)
      WRITE(6,*)'     '
      WRITE(6,*)'     '
C     WRITE(6,*)' ENTER THE NAME OF THE FILE CONTAINING THE'
C     WRITE(6,*)' BOUNDARY LAYER CHARACTERISTICS . THIS FILE'
C     WRITE(6,*)' WAS PRODUCED FROM A RIM DATA BASE CONTAINING'
C     WRITE(6,*)' PREDICTIONS FROM THE WHITFIELD PROGRAM.'
C     WRITE(6,*)'  FILE NAME (13 CHARS. MAX.) ??'
C     READ(5,2) RMFILE
      RMFILE(1:8)='LINK.DAT'
2     FORMAT(A13)
      WRITE(6,*)'   '
      WRITE(6,*)'   '
      WRITE(6,*)' ENTER THE NAME OF THE FILE CONTAINING THE'
      WRITE(6,*)' ORIGINAL PAN AIR PROBLEM WITHOUT VISCOUS'
      WRITE(6,*)' EFFECTS . '
      WRITE(6,*)'     FILE NAME ? (20 CHARS. MAX.)'
      WRITE(6,*)'   '
      READ(5,3) PAFILE
3     FORMAT(A20)
      WRITE(6,*)'   '
      WRITE(6,*)' ENTER THE FILE NAME FOR THE NEW PAN AIR PROBLEM'
      WRITE(6,*)' WITH VISCOUS EFFECTS SIMULATED TO BE GENERATED'
      WRITE(6,*)' BY THIS PROGRAM .'
      WRITE(6,*)'   '
      WRITE(6,*)' ENTER FILE NAME (20 CHARS. MAX.)'
      READ(5,3) NEWFILE
      OPEN(UNIT=2,FILE='LINK.DAT',STATUS='OLD')
      OPEN(UNIT=3,FILE=PAFILE,STATUS='OLD')
      OPEN(UNIT=4,FILE=NEWFILE,STATUS='NEW')
      NNCOUNT=0
C
C  READ 3 TITLE CARDS IN RIM FILE
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C
      DO I=1,3
      READ(2,4) LINE
4     FORMAT(A80)
      END DO
5     CONTINUE
C NOW READ THE FIRST NETWORK FOR WHICH WE VISCOUS CORRECTIONS
C TO MAKE
C
      IOS=0
      CALL READ2(IOS,PNTS,NN,X,Y,Z,DELSTAR,IRN,ICN,SLP)
C
C NOW CHECK TO SEE IF THIS IS THE LAST NETWORK FOR WHICH
C CORRECTIONS ARE TO BE MADE
C
      IF(IOS.NE.0)THEN
      CALL LASTRITE(LINE)
      GO TO 900
      END IF
20    CONTINUE
C
C  NOW READ A NETWORK FROM THE ORIGINAL PAN AIR PROBLEM
C
      CALL READ3(IOS,M,N,NNCOUNT,XPA,YPA,ZPA,LINE)
      IF(IOS.NE.0) GO TO 900
C
C  NOW CHECK TO SEE IF NETWORK WITH BOUNDARY LAYER PREDICTIONS
C  COINCIDES WITH NETWORK FROM ORIGINAL PAN AIR PROBLEM.
C
      IF(NN(1).EQ.NNCOUNT) GO TO 30
C
C  THE NETWORKS DO NOT COINCIDE SO READ NEXT NETWORK FROM
C  PAN AIR PROBLEM.
C
      GO TO 20
30    CONTINUE
C
C THE NETWORKS COINCIDE SO NOW WRITE OUT THE NEW PAN AIR PROBLEM
C  INCLUDING BOUNDARY CONDITIONS.
C
C  FIRST IDENTIFY THE TYPE (UPPER OR LOWER)
C  OF BOUNDARY CONDITION IN THE ORIGINAL PAN AIR PROBLEM
C
      CALL IDBC(BC,LINE)
C
C  NOW WRITE OUT THE NEW BOUNDARY CONDITIONS.
C
      CALL RITEBC(BC,SLP,M,N,PNTS,XPA,YPA,ZPA,X,Y,Z,S1,S2,S3,S4)
      GO TO 5
900   CONTINUE
      STOP
      END
C
C
C
      SUBROUTINE READ2(IOS,PNTS,NN,X,Y,Z,DELSTAR,IRN,ICN,SLP)
      DIMENSION NN(500),X(500),Y(500),Z(500),S(500),DELSTAR(500)
```

```
      DIMENSION IRN(500),ICN(500),SLP(500)
C
C  THIS SUBROUTINE READS A NETWORK FROM THE FILE CONTAINING THE
C  PREDICTED BOUNDARY LAYER CHARACTERISTICS FROM WHITFIELD PROGRAM
C
      PNTS=0
      I=1
      READ(2,600,IOSTAT=IOS) NN(I),X(I),Y(I),Z(I),S(I),DELSTAR(I),
     2 IRN(I),ICN(I),SLP(I)
600   FORMAT(I9,5F10.5,2I10,1F10.5)
      IF(IOS.NE.0)THEN
      PNTS=I-1
      RETURN
      END IF
      I=I+1
      READ(2,600,IOSTAT=IOS) NN(I),X(I),Y(I),Z(I),S(I),DELSTAR(I),
     2 IRN(I),ICN(I),SLP(I)
      IF(IOS.NE.0)THEN
      PNTS=1
      RETURN
      END IF
      DO WHILE(NN(I).EQ.NN(I-1))
      I=I+1
      read(2,600,END=602) nn(i),x(i),y(i),z(i),s(i),delstar(i)
     2  ,IRN(I),ICN(I),SLP(I)
      END DO
602   CONTINUE
      PNTS=I-1
      BACKSPACE(UNIT=2)
      IOS=0
      RETURN
      END
      SUBROUTINE LASTRITE(LINE)
C
C
C  ALL OF THE NETWORKS FOR WHICH A VISCOUS ANALYSIS
C  WAS MADE HAVE BEEN READ SO NOW WRITE OUT THE REST OF THE ORIGINAL PAN
C
C
      CHARACTER LINE*80
      IOS=0
614   CONTINUE
      READ(3,615,IOSTAT=IOS)LINE
615   FORMAT(A80)
      WRITE(4,615) LINE
      IF(IOS.NE.0) GO TO 618
      GO TO 614
618   CONTINUE
      RETURN
      END
C
C
      SUBROUTINE READ3(IOS,M,N,NNCOUNT,XPA,YPA,ZPA,LINE)
C
C
C  THIS SUBROUTINE READS THE ORIGINAL PAN AIR PROBLEM
C  ( ONE NETWORK AT A TIME ) AND WRITES THE NETWORK OUT TO THE
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C   NEW PAN AIR PROBLEM
C
C
      CHARACTER LINE*80
      DIMENSION XPA(20,20),YPA(20,20),ZPA(20,20)
C
C   FIND THE NEXT NETWORK AND WRITE OUT TO THE NEW FILE ALL
C   LINES BEFORE THE NETWORK DEFINITION BEGINS.
C
      IOS=0
639   CONTINUE
      READ(3,640,IOSTAT=IOS) LINE
640   FORMAT(A80)
      IF(IOS.NE.0) THEN
      RETURN
      END IF
      WRITE(4,641) LINE
641   FORMAT(A80)
C
C   FIND THE LINE WITH THE INITIAL CHARACTERS 'NETWORK'
C
      DO I=1,10
      J=I+3
      IF(LINE(I:J).EQ.'NETW') GO TO 642
      END DO
C
C   TRY NEXT LINE
C
      GO TO 639
642   CONTINUE
C
C   WE HAVE FOUND THE BEGINNING OF A NETWORK
C
C
C   NOW LOCATE THE '=' SIGN
C
      DO K=J,55
      IF(LINE(K:K).EQ.'=') GO TO 643
      END DO
643   CONTINUE
C
C   NOW FIND THE NETWORK NAME
C
      N1=K+1
      DO J=K,50
      IF(LINE(J:J).NE.' ') GO TO 644
      N1=N1+1
      END DO
C
C   FIND END OF NETWORK NAME
C
644   CONTINUE
      N2=N1
      DO I=N1,55
      IF(LINE(I:I).EQ.' ')THEN
      N2=I-1
      GO TO 645
```

```
      END IF
      END DO
645   CONTINUE
C
C  NOW FIND THE DIMENSIONS (MXN) OF THE NETWORK
C
C  FIRST FIND M
C
      M=0
      DO I=N2+1,80
      IF(LINE(I:I).NE.' ') THEN
      READ(LINE(I:I),646,ERR=639)M
646   FORMAT(I1)
      N3=I+1
      IF(LINE(N3:N3).NE.' ')THEN
      READ(LINE(I:I+1),647,ERR=639)M
647   FORMAT(I2)
      N3=I+2
      END IF
      END IF
      IF(M.NE.0)GO TO 648
      END DO
648   CONTINUE
      IF(I.EQ.80) GO TO 639
C
C  NOW FIND N
C
      N=0
      DO I=N3,80
      IF(LINE(I:I).NE.' ')THEN
      READ(LINE(I:I),650,ERR=639) N
650   FORMAT(I1)
      IF(LINE(I+1:I+1).NE.' ') THEN
      READ(LINE(I:I+1),651,ERR=639) N
651   FORMAT(I2)
      END IF
      END IF
      IF(N.NE.0) GO TO 652
      END DO
652   CONTINUE
      IF(I.EQ.80) GO TO 639
C
C  NOW READ AND WRITE THE COORDINATES OF THE NETWORK

      NNCOUNT=NNCOUNT+1
789   FORMAT(5X,'M= ',I3,10X,'N= ',I3)
      DO JJ=1,N
      DO KK=1,M
      READ(3,*) XPA(JJ,KK),YPA(JJ,KK),ZPA(JJ,KK)
      WRITE(4,790) XPA(JJ,KK),YPA(JJ,KK),ZPA(JJ,KK)
790   FORMAT(3F20.5)
      END DO
      END DO
      IOS=0
      RETURN
      END
C
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C
      SUBROUTINE BCINTERP(SLP,M,N,XPA,YPA,ZPA,X,Y,Z,S1,S2,S3,S4)
C
C  THIS SUBROUTINE INTERPOLATES THE VELOCITIES FOR THE CENTER CONTROL
C  POINTS TO GET THE VELOCITIES FOR THE EDGE CONTROL POINTS
C
      DIMENSION S1(20),S2(20),S3(20),S4(20),X(500),Y(500),Z(500)
      DIMENSION SLP(500),XPA(20,20),YPA(20,20),ZPA(20,20)
CV
C  FIRST EDGE
C
      L1-(M-1)*(N-1)
      L2=M-1
      L3=N-1
      DO L=1,L3
C
C  INTERPOLATE FOR ORDINATES OF EDGE 1 CONTROL POINTS
C
      XCP=(XPA(L,1)+XPA(L+1,1))/2.0
      YCP=(YPA(L,1)+YPA(L+1,1))/2.0
      ZCP=(ZPA(L,1)+ZPA(L+1,1))/2.0
      KKK=2+(L-1)*(M-1)
      DELS=SLP(KKK)-SLP(KKK-1)
      DELX=X(KKK)-X(KKK-1)
      VALUE=SLP(KKK-1)+((DELS/DELX)*(XCP-X(KKK-1)))
      S1(L)=VALUE
      IF(VALUE.LT.0.0) S1(L)=0.0
      END DO
C
C  SECOND EDGE
C
      DO L=1,L2
      XCP=(XPA(N,L)+XPA(N,L+1))/2.0
      YCP=(YPA(N,L)+YPA(N,L+1))/2.0
      ZCP=(ZPA(N,L)+ZPA(N,L+1))/2.0
      KKK=(M-1)*(N-1)-(M-1)+L
      JJJ=(M-1)*(N-1)-2*(M-1)+L
      DELS=SLP(JJJ)-SLP(KKK)
      DELY=Y(JJJ)-Y(KKK)
      VALUE=SLP(KKK)+((DELS/DELY)*(YCP-Y(KKK)))
      S2(L)=VALUE
      IF(VALUE.LT.0.0) S2(L)=0.0
      END DO
C
C  EDGE 3
C
      DO L=1,L3
      L4=N+1-L
      XCP=(XPA(L4,M)+XPA(L4-1,M))/2.0
      YCP=(YPA(L4,M)+YPA(L4-1,M))/2.0
      ZCP=(ZPA(L4,M)+ZPA(L4-1,M))/2.0
      KKK=(M-1)*(N-1)-(L-1)*(M-1)
      DELS=SLP(KKK-1)-SLP(KKK)
      DELX=X(KKK-1)-X(KKK)
      VALUE=SLP(KKK)+((DELS/DELX)*(XCP-X(KKK)))
      S3(L)=VALUE
      IF(VALUE.LT.0.0) S3(L)=0.0
```

```
      END DO
C
C  EDGE 4
C
      DO L=1,L2
      L5=M+1-L
      XCP=(XPA(1,L5)+XPA(1,L5-1))/2.0
      YCP=(YPA(1,L5)+YPA(1,L5-1))/2.0
      ZCP=(ZPA(1,L5)+ZPA(1,L5-1))/2.0
      KKK=M-L
      JJJ=2*(M-1)+1-L
      DELS=SLP(JJJ)-SLP(KKK)
      DELY=Y(JJJ)-Y(KKK)
      VALUE=SLP(KKK)+((DELS/DELY)*(YCP-Y(KKK)))
      S4(L)=VALUE
      IF(VALUE.LT.0.0) S4(L)=0.0
      END DO
      RETURN
      END
C
C
C
      SUBROUTINE IDBC(BC,LINE)
C
C  THIS SUBROUTINE FINDS THE ORIGINAL BOUNDARY CONDITION AND DETERMINES
C  WHETHER IT WAS A UPPER OR LOWER SUBCLASS
C
      CHARACTER LINE*80,BC*4
699   CONTINUE
      READ(3,700,IOSTAT=IOS) LINE
      IF(IOS.NE.0) RETURN
700   FORMAT(A80)
C
C  CHECK FOR 'BOUN' IDENTIFIER
C
      DO I=1,10
      J=I+3
      IF(LINE(I:J).EQ.'BOUN') GO TO 705
      END DO
C
C  THIS LINE DOES NOT SPECIFY THE BOUNDARY CONDITION SO
C  WRITE THIS LINE TO THE NEW FILE AND GO TO NEXT LINE
C
      WRITE(4,700) LINE
      GO TO 699
705   CONTINUE
C
C  THIS LINE CONTAINS THE BC SO SORT OUT THE REST
C  OF THE LINE
C
C  FIND THE '=' SIGN
C
      DO K=J,50
      IF(LINE(K:K).EQ.'=') GO TO 710
      END DO
710   CONTINUE
C
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C  NOW LOOK FOR 'UPPER' OR 'LOWER'
      DO IK=K,76
      IJ=IK+3
      IF(LINE(IK:IJ).EQ.'UPPE') GO TO 725
      IF(LINE(IK:IJ).EQ.'LOWE') GO TO 725
      END DO
      WRITE(6,720)
720   FORMAT(//,20X,' THE BOUNDARY CONDITION MUST BE CLASS 1,',/,
     2  20X,'SSUBCLASS UPPER OR LOWER PRIOR TO THE VISCOUS ANALYSIS',
     3  /,20X,'IF THE SUBCLASS INDEX WAS USED INSTEAD OF THE KEYWORD',
     4 /,20X,'THEN CHANGE TOO THE KEYWORD DESIGNATION.')
725   READ(LINE(IK:IJ),726) BC
726   FORMAT(A4)
C
C  THE BOUNDARY CONDITION SUBCLASS HAS NOW BEEN IDENTIFIED.
C  NOW WRITE OUT ANY ADDITIONAL LINES WHICH EXISTED IN THE
C  ORIGINAL PAN AIR PROBLEM BETWEEN THE BC AND THE NEXT NETWORK
C
727   CONTINUE
      READ(3,700,END=740) LINE
C
C  NOW CHECK FOR NETWORK DESIGNATION
C
      DO II=1,20
      JJ=II+3
      IF(LINE(II:JJ).EQ.'NETW') THEN
      BACKSPACE(UNIT=3)
      GO TO 740
      END IF
      END DO
C
C  THIS LINE IS NOT A NEW NETWORK SO WRITE IT OUT TO THE NEW PAN AIR PRO
C
      WRITE(4,700) LINE
      GO TO 727
740   CONTINUE
      RETURN
      END
C
      SUBROUTINE RITEBC(BC,SLP,M,N,PNTS,XPA,YPA,ZPA,X,Y,Z,S1,S2,S3,S4)
C
C
C  THIS SUBROUTINE WRITES OUT THE NEW BOUNDARY CONDITIONS FOR THE VISCOU
C
      CHARACTER BC*4
      DIMENSION XPA(20,20),YPA(20,20),ZPA(20,20),X(500),Y(500),Z(500),
     2 SLP(500)
      DIMENSION S1(20),S2(20),S3(20),S4(20)
      WRITE(4,755) BC
755   FORMAT('BOUNDARY CONDITION = 2,',A4)
      WRITE(4,756)
756   FORMAT('SPECIFIED FLOW',/,'TERM = 1')
      WRITE(4,757)
757   FORMAT('POINTS = CENTER ')
      WRITE(4,758)
758   FORMAT('*/  OUTFLOW VELOCITIES FOR PANEL CENTER POINTS')
      ILIM=(M-1)*(N-1)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      IF(PNTS.NE.ILIM) THEN
      WRITE(6,*)' '
      WRITE(6,*)' CAUTION : THE NUMBER OF BOUNDARY CONDITIONS PRODUCED '
      WRITE(6,*)' FROM THE WHITFIELD PROGRAM DOES NOT MATCH THE NUMBER'
      WRITE(6,*)' OF PANELS READ FROM THE ORIGINAL PAN AIR PROBLEM. '
      WRITE(6,*)' CHECK NEW PROBLEM BEFORE EXECUTING.'
      WRITE(6,*)'  '
      WRITE(6,759) M,N,PNTS
759   FORMAT(6X,'M= ',I3,5X,'N= ',I3,5X,'PNTS= ',I3)
      END IF
      IF(BC(1:4).EQ.'LOWE') THEN
      DO I=1,PNTS
      SLP(I)=-SLP(I)
      END DO
      END IF
      WRITE(4,760) (SLP(I), I=1,PNTS)
760   FORMAT(6(F10.6,2X))
      WRITE(4,761)
761   FORMAT(3('*/',/),'POINTS = EDGE')
      WRITE(4,762)
762   FORMAT('*/  BEGIN VELOCITIES FOR EDGE 1')
C
C  NOW INTERPOLATE VELOCITIES FOR CENTER CONTROL POINTS TO GET VELOCITIE
C  FOR EDGE CONTROL POINTS
C
      CALL BCINTERP(SLP,M,N,XPA,YPA,ZPA,X,Y,Z,S1,S2,S3,S4)
C
C   WRITE OUT EDGE VELOCITIES AND CALCULATE AVG. VELOCITIES FOR ADDITION
C
      A1=0.0
      AI=0.0
      DO I=1,N-1
      AI=AI+1.0
      WRITE(4,763) S1(I)
      A1=A1+S1(I)
763   FORMAT(2X,F12.6)
      END DO
      A1=A1/AI
      WRITE(4,765)
765   FORMAT('*/',/,'*/')
      WRITE(4,766)
766   FORMAT('*/  BEGIN EDGE VELOCITIES FOR SIDE 2')
      A2=0.0
      AI=0.0
      DO I=1,M-1
      AI=AI+1.0
      WRITE(4,763) S2(I)
      A2=A2+S2(I)
      END DO
      A2=A2/AI
      WRITE(4,765)
      WRITE(4,767)
767   FORMAT('*/  BEGIN EDGE VELOCITIES FOR SIDE 3')
      A3=0.0
      AI=0.0
      DO I=1,N-1
      AI=AI+1.0
```

```
      WRITE(4,763) S3(I)
      A3=A3+S3(I)
      END DO
      A3=A3/AI
      WRITE(4,765)
      WRITE(4,768)
768   FORMAT('*/  BEGIN EDGE VELOCITIES FOR SIDE 4')
      A4=0.0
      AI=0.0
      DO I=1,M-1
      AI=AI+1.0
      WRITE(4,763) S4(I)
      A4=A4+S4(I)
      END DO
      A4=A4/AI
C
C  NOW WRITE ADDITIONAL CONTROL POINTS
C
      WRITE(4,765)
      WRITE(4,769)
769   FORMAT('POINTS = ADDITIONAL')
      WRITE(4,770) A1
770   FORMAT(2X,'( 1 ) = ',F12.6)
      WRITE(4,771) A2
771   FORMAT(2X,'( 2 ) = ',F12.6)
      WRITE(4,772) A3
772   FORMAT(2X,'( 3 ) = ',F12.6)
      WRITE(4,773) A4
773   FORMAT(2X,'( 4 ) = ',F12.6)
      WRITE(4,765)
      RETURN
      END
```

C.3 Sample Run of Viscous Simulation Module

The simplified wing-body problem shown in Figure C-1 was analyzed to demonstrate the Viscous Simulation Module. The inviscid flow field properties for this configuration were first determined using PAN AIR. The PAN AIR problem for this configuration is presented in Section C.3.2. The inviscid flow properties were then used to calculate the boundary layer on the wing upper and lower surfaces. A new PAN AIR problem with the wing boundary layer simulated is presented in Section C.3.3. This process was accomplished completely within the Viscous Simulation Module with the inputs and execution shown in Section C.3.1.

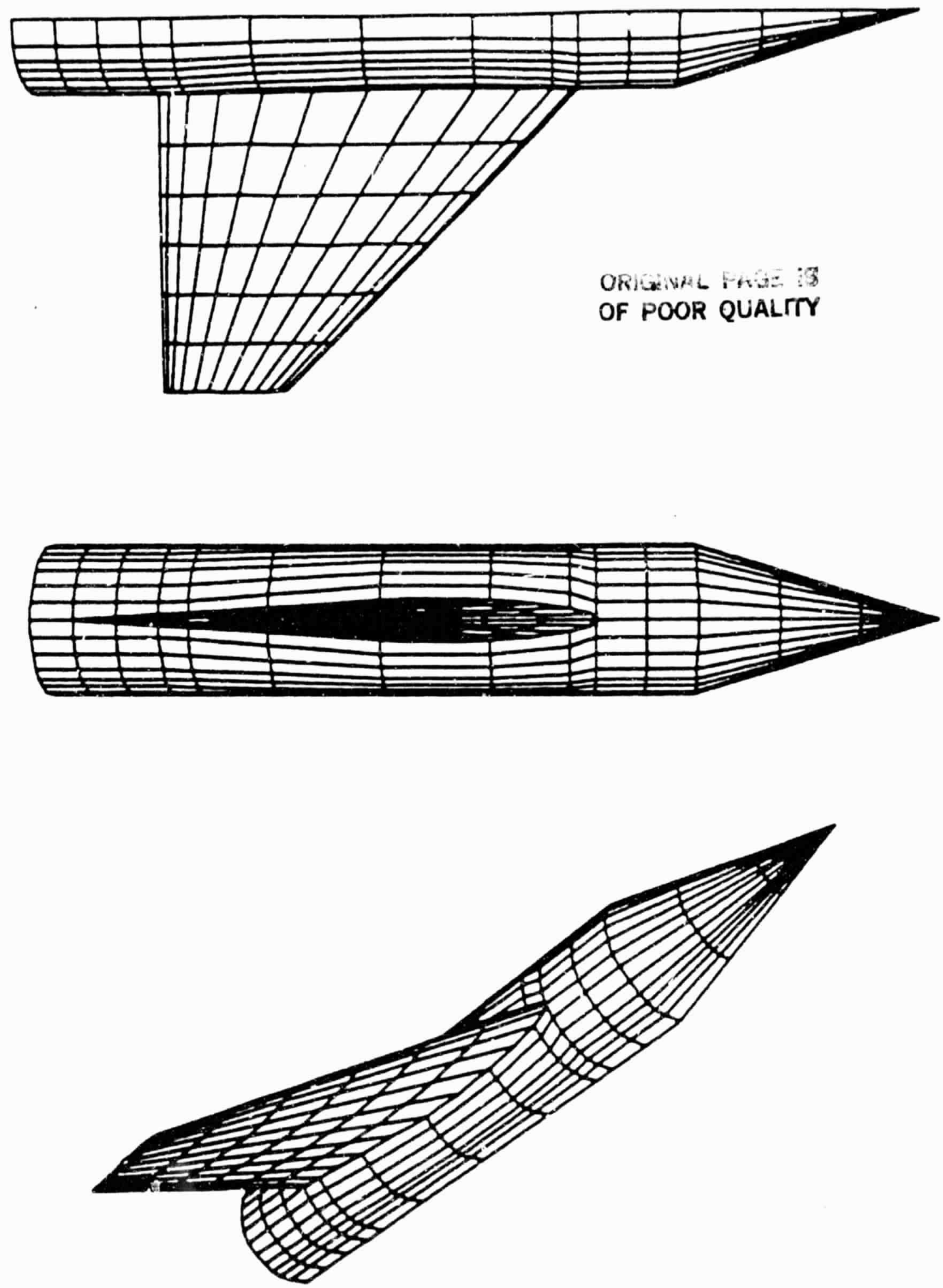


Figure C-1 Wing-Body Intersection Problem

C.3.1 Sample Execution of Viscous Simulation Module

The execution of the Viscous Simulation Module which converted the PAN AIR problem presented in Section C.3.2 to the new PAN AIR problem presented in Section C.3.3 containing specified boundary conditions is illustrated in the following pages.

```
$ @VISC

 EXECUTION OF THE VISCOUS EFFECTS MODULE IS BEGINNING.


 FIRST DELETE PREVIOUS DATA BASE GENERATED BY EXECUTION OF THIS MODULE.

STA0:[BHATELEY.PANAIR]NETWKS1.DAT;1, delete? (Y or N): Y
STA0:[BHATELEY.PANAIR]NETWKS2.DAT;1, delete? (Y or N): Y
STA0:[BHATELEY.PANAIR]NETWKS3.DAT;1, delete? (Y or N): Y

 THE PROGRAM EDITTEN WHICH READS A PAN AIR TAPE 10 WILL NOW BE EXECUTED.
 THIS PROGRAM WILL REQUEST THE NAME OF THE TAPE 10 FILE WHICH WAS GENERATED BY PAN AIR .


THE FILES PATEMP.COM AND RIMINFO.TMP WILL BE GENERATED BY THE EDITTEN PROGRAM .

PATEMP.COM WILL BE EXECUTED BY THIS COMMAND FILE SO IGNORE THE INSTRUCTIONS TO USE THE FORK COMMAND .

 BOTH THE DATA BASE NAME AND RELATIONS NAME TO BE USED IN RIM WILL BE SET TO 'NETWKS' .
 THE DATA BASE GENERATED BY RIM WILL CONSIST OF THE FILES NETWKS1.DAT , NETWKS2.DAT AND NETWKS3.DAT.
```

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

```
Program EDITTEN.

Purpose:  To read in a PAN AIR TAPE10 and change its form so RIM can read it.

 Enter name of TAPE10 input file
MANHBCJUP.OU
  Working now......

By the way, this may take a while...

The files PATEMP.COM and RIMINFO.TMP have been written.

RIMINFO.TMP contains information about your RIM Data Base.  You should print it out.

PATEMP.COM is a command file that can be executed best by using FORK.
Just do the following:

   1. Type in FORK<cr> (you will get a "Commands:" prompt)
   2. Type in $ @PATEMP<cr>
   3. Type in <cr>

Just be sure NOT to log off before FORK tells you your job is finished.

 PATEMP.COM WILL BE RUN AS PART OF THIS COMMAND FILE SO
  IT WILL NOT BE NECESSARY TO USE THE FORK COMMAND .
FORTRAN STOP
 EDITTEN HAS BEEN EXECUTED.
```

```
HERE IS A LISTING OF THE PAN AIR NETWORKS
   1 NOSE-1                          2 FWD-BODY-2
   3 LWR-BODY-3                      4 UPR-BODY-4
   5 LWR-AFT-BODY-5                  6 UPR-AFT-BODY-6
   7 UPR-WING-7                      8 LWR-WING-8
   9 WING-CLOS-9                    10 EXIT-10
  11 WING-WAKE-11                   12 EXIT-UPR-WAKE-12
  13 EXIT-LWR-WAKE-13

THE FILE PATEMP.COM WHICH SETS UP A RIM DATA BASE CONTAINING PAN AIR FLOW PARAMETERS WILL NOW BE RUN.

THIS MAY TAKE AWHILE , PARTICULARLY IF THE PAN AIR PROBLEM HAS NUMEROUS PANELS.

THE FILE LABEL.TMP CONTAINING COMMENTS FROM EXECUTION OF RIM WILL BE GENERATED.
IT WILL LATER BE PURGED AS IT DOES NOT CONTAIN ANY USEFUL INFO.

Previous logical name assignment replaced
FORTRAN STOP
Previous logical name assignment replaced

PATEMP.COM WHICH SETS UP THE DATA BASE NAMED NETWKS HAS BEEN EXECUTED .

HERE ARE THE NETWORKS AGAIN ; YOU NEED TO KNOW THE ID NUMBERS
   1 NOSE-1                          2 FWD-BODY-2
   3 LWR-BODY-3                      4 UPR-BODY-4
   5 LWR-AFT-BODY-5                  6 UPR-AFT-BODY-6
   7 UPR-WING-7                      8 LWR-WING-8
   9 WING-CLOS-9                    10 EXIT-10
  11 WING-WAKE-11                   12 EXIT-UPR-WAKE-12
  13 EXIT-LWR-WAKE-13

THE PROGRAM SELECT WILL NOW BE EXECUTED .
THIS PROGRAM GENERATES A COMMAND FILE (RUNRIM.COM) TO SELECT FROM THE RIM DATA BASE (NETWKS)
  THE FLOW PARAMETERS AND GEOMETRY REQUIRED BY THE WHITFIELD BOUNDARY LAYER PROGRAM .

 ENTER THE NUMBER OF NETWORKS ON THE UPPER
 SURFACE THAT YOU WANT TO COMBINE FOR A
 VISCOUS ANALYSIS  ( ENTER 1 IF THE UPPER
SURFACE IS COMPOSED OF ONLY 1 NETWORK)
1
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
  ENTER THE NUMBER OF NETWORKS ON THE LOWER
  SURFACE THAT YOU WANT TO COMBINE FOR A
  VISCOUS (INTERGER FORMAT). ENTER 1 IF
  THE LOWER SURFACE IS COMPOSED OF ONLY 1 NETWORK
1
  NOW ENTER THE IDENTIFICATION NUMBERS OF THE
  UPPER SURFACE NETWORKS REQUIRED FOR THE VISCOUS
  ANALYSIS. ENTER THEM IN INTERGER FORM , ON ONE LINE
  SEPERATED BY COMMAS AS IN THE FOLLOWING EXAMPLE
  1,2,3,4 ETC.
7
  NOW ENTER THE IDENTIFICATION NUMBERS OF
  THE LOWER SURFACE NETWORKS. ENTER THEM
  IN INTERGER FORM , ON ONE LINE , SEPERATED
  BY COMMAS AS IN THE FOLLOWING EXAMPLE 5,6,7,8
8
FORTRAN STOP


 THE SELECT PROGRAM HAS BEEN EXECUTED .

 RIM WILL NOW BE RUN TO GENERATE THE FILES WINGU.DAT AND WINGL.DAT WHICH CONTAIN THE WING PARAMETERS
  REQUIRED BY THE WHITFIELD PROGRAM.

Previous logical name assignment replaced
FORTRAN STOP
Previous logical name assignment replaced


 RIM HAS BEEN EXECUTED .


 THE FILES WINGU.DAT AND WINGL.DAT WILL NOW BE USED AS INPUT TO THE PROGRAM WHITIN.
 WHITIN PREPARES A INPUT FILE FOR THE WHITFIELD BOUNDARY LAYER PROGRAM.


   WHITIN IS NOW BEING EXECUTED .
```


C-4

```
  INPUT THE DESIRED TOLERANCE DIMENSION FOR
  DETERMINING STATION CUTS ALONG SURFACE

      TOLERANCE = ?
3.0
 ENTER FREE STREAM MACH NUMBER FOR BOUNDARY
 LAYER ANALYSIS
         MACH = ???
0.60
ENTER UNIT REYNOLDS NUMBER PER MILLION
    REN = ????? (PER 10E6)
5.0
  THREE FILES HAVE NOW BEEN GENERATED :
       1 : UPPER.DAT
       2 : LOWER.DAT
       3 : STAG.DAT
  UPPER.DAT AND LOWER.DAT ARE IN THE FORMAT
  REQUIRED FOR INPUT INTO THE WHITFIELD
  BOUNDARY LAYER PROGRAM
  STAG.DAT CONTAINS THE PANEL POINTS DEFINED
  AS STAGNATION POINTS ; THIS FILE WILL BE
  USED LATER TO GENERATE THE BOUNDARY CONDITIONS
  AFTER THE OTHER TWO FILES HAVE BEEN ANALYZED
  IN THE WHITFIELD PROGRAM.
FORTRAN STOP


 THE PROGRAM WHITIN HAS BEEN EXECUTED.


 THE FILES UPPER.DAT AND LOWER.DAT HAVE BEEN GENERATED FOR INPUT TO THE WHITFIELD PROGRAM.

 THE WHITFIELD PROGRAM IS BEING RUN FOR THE UPPER SURFACE.

FORTRAN STOP
Previous logical name assignment replaced
Previous logical name assignment replaced

 THE WHITFIELD PROGRAM IS BEING RUN FOR THE LOWER SURFACE.
```

```
FORTRAN STOP
 EXECUTION OF THE WHITFIELD PROGRAM IS COMPLETE .

 THE FILES UPROUT.DAT AND LOWOUT.DAT HAVE BEEN GENERATED.
 THESE FILES WILL BE INPUT TO THE WHITOUT PROGRAM.

 THE WHITOUT PROGRAM WILL PRODUCE A COMMAND FILE (REORDER.COM) .
 REORDER.COM WILL LOAD A RIM DATA BASE (ORDER); THIS IS NECESSARY TO SORT
 THE WHITFIELD OUTPUT BACK INTO A FORMAT SUITABLE FOR PAN AIR.

FORTRAN STOP
 THE WHITOUT PROGRAM HAS BEEN EXECUTED .

 FIRST DELETE THE PREVIOUS DATA BASE GENERATED:

STAO:[BHATELEY.PANAIR]ORDER1.DAT;1, delete? (Y or N): Y
STAO:[BHATELEY.PANAIR]ORDER2.DAT;1, delete? (Y or N): Y
STAO:[BHATELEY.PANAIR]ORDER3.DAT;1, delete? (Y or N): Y

 THE COMMAND FILE REORDER.COM IS NOW BEING EXECUTED .

Previous logical name assignment replaced
FORTRAN STOP
Previous logical name assignment replaced


 THE DATA BASE ORDER HAS BEEN GENERATED.

 THE PROGRAM SORT WILL NOW BE EXECUTED , THIS PROGRAM OPENS THE DATA BASE ORDER
 AND SORTS THE OUTPUT FROM THE WHITFIELD PROGRAM INTO THE ORDER REQUIRED FOR
  THE NEW PAN AIR PROBLEM WHICH HAS VISCOUS EFFECTS SIMULATED .


FORTRAN STOP
 THE PROGRAM SORT HAS BEEN EXECUTED ; THE FILE RIMSORT.COM HAS BEEN GENERATED

 THE COMMAND FILE RIMSORT.COM WILL NOW BE RUN .
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
 THIS FILE WILL OPEN THE DATA BASE NAMED ORDER AND REARRANGE
 THE DATA PARAMETERS INTO A FORMAT COMPATABLE WITH PAN AIR .
 RIMSORT IS NOW EXECUTING .
Previous logical name assignment replaced
FORTRAN STOP
Previous logical name assignment replaced

RIMSORT IS FINISHED . THE FILE LINK.DAT HAS BEEN GENERATED .

 THE PROGRAM LINK WILL NOW BE RUN . THIS PROGRAM WILL WRITE THE NEW PAN AIR
 PROBLEM WHICH CONTAINS THE BOUNDARY CONDITION SIMULATING VISCOUS EFFECTS .



 ENTER THE NAME OF THE FILE CONTAINING THE
 ORIGINAL PAN AIR PROBLEM WITHOUT VISCOUS
 EFFECTS .
    FILE NAME ? (20 CHARS. MAX.)

RIP4.JOB

 ENTER THE FILE NAME FOR THE NEW PAN AIR PROBLEM
 WITH VISCOUS EFFECTS SIMULATED TO BE GENERATED
 BY THIS PROGRAM .

 ENTER FILE NAME (20 CHARS. MAX.)
RIP4VISC.JOB
FORTRAN STOP


 THE PROGRAM LINK HAS BEEN EXECUTED .



VISC.COM HAS BEEN EXECUTED
$
```

ORIGINAL PAGE IS
OF POOR QUALITY

C.5.2 Original PAN AIR Problem

The input problem for the original inviscid PAN AIR run is presented in the following pages.

```
MANNHW,PN,T1100,CM154000,7D2,7L2,RS.
ACCOUNT(FHASB,T5520)
DISPOSE,OUTPUT,ST=ARS1XD,*PR.
COMMENT. RUN THE PRODUCTION VERSION OF PANAIR
COMMENT. STORED AS L12,1JRUNPANAIR.JCL ON CAFAIR.......................
COMMENT.
COPYSP(INPUT,OUTPUT)
REWIND(INPUT)
SETNAME(FHASBGD)
MOUNT(VSN=D0111D)
REQUEST,TAPE9,SN=FHASBGD,*PF.
REQUEST,TAPE10,SN=FHASBGD,*PF.
REQUEST,TAPE11,SN=FHASBGD,*PF.
AUDIT(ID=FHAICB,AI=P)
COMMENT. PURGE ANY OLD DATASETS THAT MIGHT BE THERE
EXIT,U.
COMMENT. ATTACH AND EXECUTE THE MEC MODULE
ATTACH(MEC,ID=PANAIR)
MEC.
SUMMARY.
RETURN(MEC)
COMMENT. ATTACH AND EXECUTE THE DIP MODULE
ATTACH(DIP,ID=PANAIR)
DIP(INPUT,PL=100000)
SUMMARY.
RETURN(DIP)
COMMENT. ATTACH AND EXECUTE THE DQG MODULE
ATTACH(DQG,ID=PANAIR)
DQG(PL=100000)
SUMMARY.
RETURN(DQG)
COMMENT. ATTACH AND EXECUTE THE MAG MODULE
ATTACH(MAG,ID=PANAIR)
MAG.
SUMMARY.
RETURN,MAG.
COMMENT. ATTACH AND EXECUTE THE RMS MODULE
ATTACH(RMS,ID=PANAIR)
RMS.
SUMMARY.
RETURN(RMS)
COMMENT. ATTACH AND EXECUTE THE RHS MODULE
ATTACH(RHS,ID=PANAIR)
RHS.
SUMMARY.
RETURN(RHS)
COMMENT. ATTACH AND EXECUTE THE MDG MODULE
ATTACH(MDG,ID=PANAIR)
MDG.
SUMMARY.
RETURN(MDG)
COMMENT. ATTACH AND EXECUTE THE PDP MODULE
ATTACH(PDP,ID=PANAIR)
PDP(PL=40000)
SUMMARY.
RETURN(PDP)
COMMENT. ATTACH AND EXECUTE THE CDP MODULE
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
ATTACH(CDP,CDP,ID=PANAIR)
CDP.
SUMMARY.
RETURN(CDP)
COMMENT. ATTACH AND EXECUTE THE PPP MODULE
ATTACH(PPP,ID=PANAIR)
PPP.
SUMMARY.
RETURN(PPP)
SUMMARY.
EXIT,U.
REWIND(TAPE9,TAPE10,TAPE11)
COPYSBF(TAPE9,OUTPUT)
COPYSBF(TAPE10,OUTPUT)
COPYSBF(TAPE11,OUTPUT)
EXIT,U.
REWIND(INPUT)
CATALOG,TAPE9,PPP9F,SN=FHASBGD,ID=FHAICB.
CATALOG,TAPE10,PPP10F,SN=FHASBGD,ID=FHAICB.
CATALOG,TAPE11,PPP11F,SN=FHASBGF,ID=FHAICB.
AUDIT(ID=FHAICB,A1=P)
%%EOR
 PAN AIR / THIS IS THE FIRST CARD OF MEC INPUT FOR CDC 7600/SCOPE
SYSTEM PANAIR FHASBGD AMES
RID =CHECKOUT OF PANAIR METHOD FOR VSTOL ANALYSIS
DATA BASE DIRECTIVE BLOCK
 APPEND DAR TO ALL
 UID = FHAICB FOR ALL
 SET = FHASBGD FOR ALL
 MUID= PANAIR FOR ALL
 MSET= FHASBGD FOR ALL
 END DATA BASE DIRECTIVE BLOCK
CHECK DATA RUN
EXECUTION DIRECTIVE BLOCK
 FIND POTENTIAL FLOW
 END EXECUTION DIRECTIVE BLOCK
END PAN AIR DIRECTIVES
%%EOR
*/
*/
*/
*/ DIP INPUT - FILE RIP4.JOB
*/    WING-BODY PROBLEM FOR CHECKOUT OF VISCOUS EFFECTS MODULE
*/
*/
*/
*/
BEGIN GLOBAL DATA
 PID=MODEL TO CHECK OUT VISCOUS EFFECTS MODULE
UID = BHATELEY/HOWELL GENERAL DYNAMICS 817 732 4811
CONFIGURATION = FIRST          /ONE PLANES OF SYMMETRY.
MACH = 0.600, CALPHA = 4.0, CBETA = 0.0/ RECORD 65
ALPHA = 4.0
SID =ZERO-DEGREE                                /66
TOLERANCE = 0.1                                          /67
SURFACE SELECTION = UPPER                                /68
SELECTION OF VELOCITY COMPUTATION= BOUNDARY-CONDITION   /69
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
PRESSURE COEFFICIENT RULES=ISENTROPIC,SECOND-ORDER       7G12
*/
CHECKOUT PRINTS = DIF,1,2,3,    DQG,1,4,5
BEGIN NETWORK DATA
*/  THESE ARE THE NETWORKS FOR WING-BODY MODEL
*/
 NETWORK=NOSE-1    15     4
              0.00000              0.00000              0.00000
              0.00000              0.00000              0.00000
              0.00000              0.00000              0.00000
              0.00000              0.00000              0.00000
              0.00000              0.00000              0.00000
              0.00000              0.00000              0.00000
              0.00000              0.00000              0.00000
              0.00000              0.00000              0.00000
              0.00000              0.00000              0.00000
              0.00000              0.00000              0.00000
              0.00000              0.00000              0.00000
              0.00000              0.00000              0.00000
              0.00000              0.00000              0.00000
              0.00000              0.00000              0.00000
              0.00000              0.00000              0.00000
             33.33334              0.00000            -10.00000
             33.33334              3.45899             -9.37681
             33.33334              5.34456             -8.61779
             33.33334              6.50164             -7.65274
             33.33334              7.68819             -6.34813
             33.33334              8.83037             -4.42559
             33.33334              9.67205             -2.17988
             33.33334             10.00000              0.00000
             33.33334              9.67205              2.17988
             33.33334              8.83037              4.42559
             33.33334              7.68818              6.34813
             33.33334              6.50164              7.65275
             33.33334              5.34455              8.61779
             33.33334              3.45898              9.37681
             33.33334              0.00000             10.00000
             66.66667              0.00000            -20.00000
             66.66667              6.91797            -18.75362
             66.66667             10.68912            -17.23558
             66.66667             13.00329            -15.30549
             66.66667             15.37637            -12.69626
             66.66667             17.66074             -8.84718
             66.66667             19.34411             -4.35976
             66.66667             20.00000              0.00000
             66.66667             19.34411              4.35976
             66.66667             17.66074              8.84718
             66.66667             15.37637             12.69627
             66.66667             13.00328             15.30549
             66.66667             10.68910             17.23558
             66.66667              6.91795             18.75362
             66.66667              0.00000             20.00000
            100.00000              0.00000            -30.00000
            100.00000             10.37696            -28.15042
            100.00000             16.03367            -25.85336
            100.00000             19.50495            -22.95823
            100.00000             23.06456            -19.04439
```

ORIGINAL PAGE IS
OF POOR QUALITY

100.00000	26.49111	-13.27077
100.00000	29.01616	-6.53964
100.00000	30.00000	0.00000
100.00000	29.01616	6.53964
100.00000	26.49111	13.27077
100.00000	23.06455	19.04440
100.00000	19.50492	22.95823
100.00000	16.03366	25.85337
100.00000	10.37693	28.13043
100.00000	0.00000	30.00000

TRIANGULAR PANEL TOLERANCE = .003
BOUNDARY CONDITION = 1,LOWER
*/
*/
NETWORK=FWD-BODY-2 15 3

100.00000	0.00000	-30.00000
100.00000	10.37696	-28.13042
100.00000	16.03367	-25.85336
100.00000	19.50493	-22.95823
100.00000	23.06456	-19.04439
100.00000	26.49111	-13.27077
100.00000	29.01616	-6.53964
100.00000	30.00000	0.00000
100.00000	29.01616	6.53964
100.00000	26.49111	13.27077
100.00000	23.06455	19.04440
100.00000	19.50492	22.95823
100.00000	16.03366	25.85337
100.00000	10.37693	28.13043
100.00000	0.00000	30.00000
120.00000	0.00000	-30.00000
120.00000	10.37696	-28.13042
120.00000	16.03367	-25.85336
120.00000	19.50493	-22.95823
120.00000	23.06456	-19.04439
120.00000	26.49111	-13.27077
120.00000	29.01616	-6.53964
120.00000	30.00000	0.00000
120.00000	29.01616	6.53964
120.00000	26.49111	13.27077
120.00000	23.06455	19.04440
120.00000	19.50492	22.95823
120.00000	16.03366	25.85337
120.00000	10.37693	28.13043
120.00000	0.00000	30.00000
140.00000	0.00000	-30.00000
140.00000	10.37696	-28.13042
140.00000	16.03367	-25.85336
140.00000	19.50493	-22.95823
140.00000	23.06456	-19.04439
140.00000	26.49111	-13.27077
140.00000	29.01616	-6.53964
140.00000	30.00000	0.00000
140.00000	29.01616	6.53964
140.00000	26.49111	13.27077
140.00000	23.06455	19.04440
140.00000	19.50492	22.95823

ORIGINAL PAGE IS
OF POOR QUALITY

```
          140.00000            16.03366            25.85537
          140.00000            10.37693            28.13043
          140.00000             0.00000            30.00000
TRIANGULAR PANEL TOLERANCE = .003
BOUNDARY CONDITION = 1,LOWER
*/
*/
 NETWORK=LWR-BODY-3      8     7
          140.00000             0.00000           -30.00000
          140.00000            10.37692           -28.13042
          140.00000            16.03367           -25.85536
          140.00000            19.50493           -22.95823
          140.00000            23.06456           -19.04439
          140.00000            26.49111           -13.27077
          140.00000            29.01616            -6.53964
          140.00000            30.00000             0.00000
          150.13768             0.00000           -30.00000
          150.13768             9.43575           -28.40252
          150.13768            14.67512           -26.57329
          150.13768            18.03844           -24.32530
          150.13768            21.47990           -20.95077
          150.13768            25.04300           -16.03498
          150.13768            27.85628           -10.14802
          150.13768            29.34405            -4.38021
          180.83968             0.00000           -30.00000
          180.83968             8.68515           -28.60125
          180.83968            13.52987           -27.08412
          180.83968            16.81413           -25.32869
          180.83968            20.10723           -22.37667
          180.83968            23.72057             8.13178
          180.83968            26.65920            12.91335
          180.83968            28.56272            -7.74840
          223.97174             0.00000           -30.00000
          223.97174             8.69491           -28.59876
          223.97174            13.54501           -27.05814
          223.97174            16.83045           -25.31665
          223.97174            20.12545           -22.35871
          223.97174            23.73869           -18.10545
          223.97174            26.67559           -12.87822
          223.97174            28.57865            -7.70594
          267.20563             0.00000           -30.00000
          267.20563             9.55310           -28.37002
          267.20563            14.85005           -26.49034
          267.20563            18.22490           -24.15783
          267.20563            21.68755           -20.71791
          267.20563            25.23666           -15.69620
          267.20563            28.02673            -9.70523
          267.20563            29.42558            -3.81819
          298.05643             0.00000           -30.00000
          298.05643            10.21853           -28.17808
          298.05643            15.81293           -25.98376
          298.05643            19.26217           -23.18852
          298.05643            22.81251           -19.37369
          298.05643            26.26452           -13.74125
          298.05643            28.85916            -7.15470
          298.05643            29.88927            -0.73861
          310.00000             0.00000           -30.00000
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
          310.00000              10.37696               -28.13042
          310.00000              16.03367               -25.85336
          310.00000              19.50493               -22.95823
          310.00000              23.06456               -19.04439
          310.00000              26.49111               -13.27077
          310.00000              29.01616                -6.53964
          310.00000              30.00000                 0.00000
TRIANGULAR PANEL TOLERANCE = .003
BOUNDARY CONDITION = 1,LOWER
*/
*/
 NETWORK=UPR-BODY-4     8     7
          140.00000              30.00000                 0.00000
          140.00000              29.01616                 6.53964
          140.00000              26.49111                13.27077
          140.00000              23.06455                19.04440
          140.00000              19.50492                22.95823
          140.00000              16.03366                25.85337
          140.00000              10.37693                28.13043
          140.00000               0.00000                30.00000
          150.13760              29.34405                 4.36020
          150.13760              27.85629                10.14801
          150.13760              25.04301                16.03498
          150.13760              21.47990                20.95077
          150.13760              18.03843                24.32530
          150.13760              14.67511                26.57329
          150.13760               9.43572                28.40253
          150.13760               0.00000                30.00000
          180.83961              28.56271                 7.74840
          180.83961              26.65920                12.91336
          180.83961              23.72057                18.13179
          180.83961              20.10723                22.37668
          180.83961              16.81412                25.32869
          180.83961              13.52985                27.06412
          180.83961               8.68512                28.60125
          180.83961               0.00000                30.00000
          223.97173              28.57864                 7.70594
          223.97173              26.67559                12.87823
          223.97173              23.73868                18.10546
          223.97173              20.12545                22.35872
          223.97173              16.83044                25.31665
          223.97173              13.54499                27.05814
          223.97173               8.69489                28.59877
          223.97173               0.00000                30.00000
          267.20563              29.42558                 3.81817
          267.20563              28.02672                 9.70524
          267.20563              25.23666                15.69621
          267.20563              21.68754                20.71992
          267.20563              18.22488                24.15784
          267.20563              14.85002                26.49035
          267.20563               9.55307                28.37003
          267.20563               0.00000                30.00000
          298.05640              29.88927                 0.73601
          298.05640              28.85916                 7.15471
          298.05640              26.26452                13.74126
          298.05640              22.81250                19.37370
          298.05640              19.26215                23.18853
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
          298.05640              15.81291              25.98376
          298.05640              10.21850              28.17809
          298.05640               0.00000              30.00000
          310.00000              30.00000               0.00000
          310.00000              29.01616               6.53964
          310.00000              26.49111              13.27077
          310.00000              23.06455              19.04440
          310.00000              19.50492              22.95823
          310.00000              16.03366              25.85337
          310.00000              10.37693              28.13043
          310.00000               0.00000              30.00000
TRIANGULAR PANEL TOLERANCE = .003
BOUNDARY CONDITION = 1,LOWER
*/
*/
 NETWORK=LWR-AFT-BODY-5     8     4
          310.00000               0.00000             -30.00000
          310.00000              10.37696             -28.13042
          310.00000              16.03367             -25.85336
          310.00000              19.50493             -22.95823
          310.00000              23.06456             -19.04439
          310.00000              26.49111             -13.27077
          310.00000              29.01616              -6.53964
          310.00000              30.00000               0.00000
          326.66669               0.00000             -30.00000
          326.66669              10.37696             -28.13042
          326.66669              16.03367             -25.85336
          326.66669              19.50493             -22.95823
          326.66669              23.06456             -19.04439
          326.66669              26.49111             -13.27077
          326.66669              29.01616              -6.53964
          326.66669              30.00000               0.00000
          343.33334               0.00000             -30.00000
          343.33334              10.37696             -28.13042
          343.33334              16.03367             -25.85336
          343.33334              19.50493             -22.95823
          343.33334              23.06456             -19.04439
          343.33334              26.49111             -13.27077
          343.33334              29.01616              -6.53964
          343.33334              30.00000               0.00000
          360.00000               0.00000             -30.00000
          360.00000              10.37696             -28.13042
          360.00000              16.03367             -25.85336
          360.00000              19.50493             -22.95823
          360.00000              23.06456             -19.04439
          360.00000              26.49111             -13.27077
          360.00000              29.01616              -6.53964
          360.00000              30.00000               0.00000
TRIANGULAR PANEL TOLERANCE = .003
BOUNDARY CONDITION = 1,LOWER
*/
*/
 NETWORK=UPR-AFT-BODY-6     8     4
          310.00000              30.00000               0.00000
          310.00000              29.01616               6.53964
          310.00000              26.49111              13.27077
          310.00000              23.06455              19.04440
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
          310.00000            19.50492            22.95823
          310.00000            16.03366            25.85337
          310.00000            10.37693            28.13043
          310.00000             0.00000            30.00000
          326.66669            30.00000             0.00000
          326.66669            29.01616             6.53964
          326.66669            26.49111            13.27077
          326.66669            23.06455            19.04440
          326.66669            19.50492            22.95823
          326.66669            16.03366            25.85337
          326.66669            10.37693            28.13043
          326.66669             0.00000            30.00000
          343.33334            30.00000             0.00000
          343.33334            29.01616             6.53964
          343.33334            26.49111            13.27077
          343.33334            23.06455            19.04440
          343.33334            19.50492            22.95823
          343.33334            16 03366            25.85337
          343.33334            10.37693            28.13043
          343.33334             0.00000            30.00000
          360.00000            30.00000             0.00000
          360.00000            29.01616             6.53964
          360.00000            26.49111            13.27077
          360.00000            23.06455            19.04440
          360.00000            19.50492            22.95823
          360.00000            16.03366            25.85337
          360.00000            10.37693            28.13043
          360.00000             0.00000            30.00000
TRIANGULAR PANEL TOLERANCE = .003
BOUNDARY CONDITION = 1,LOWER
*/
*/
 NETWORK=UPR-WING-7     13      7
          260.00000           150.00000             0.00000
          260.85254           150.00000             0.43294
          263.35025           150.00000             0.80508
          267.32361           150.00000             1.13998
          272.49731           150.00000             1.38703
          278.53229           150.00000             1.49589
          285.00000           150.00000             1.38750
          291.46768           150.00000             1.08595
          297.50259           150.00000             0.70477
          302.70285           150.00000             0.35098
          306.50790           150.00000             0.13894
          308.79236           150.00000             0.04450
          310.00000           150.00000             0.00000
          240.00000           130.00000             0.00000
          241.10892           130.00000             0.72493
          244.58406           130.00000             1.39437
          250.14616           130.00000             1.99257
          257.40182           130.00000             2.43471
          265.87134           130.00000             2.62614
          274.94769           130.00000             2.42825
          284.01316           130.00000             1.89145
          292.47714           130.00000             1.22120
          299.76547           130.00000             0.60489
          305.10062           130.00000             0.23836
```

ORIGINAL PAGE IS
OF POOR QUALITY

308.30539	130.00000	0.07621
310.00000	130.00000	0.00000
220.00002	110.00002	0.00000
221.36531	110.00002	1.01693
225.81787	110.00001	1.98367
232.96872	110.00001	2.84516
242.30637	110.00001	3.48239
253.21042	110.00001	3.75640
264.89537	110.00001	3.46899
276.56866	110.00001	2.59695
287.45172	110.00001	1.73763
296.82813	110.00001	0.85881
303.69333	110.00002	0.33778
307.81839	110.00001	0.10793
310.00000	110.00001	0.00000
200.00000	90.00000	0.00000
201.62166	90.00000	1.30892
207.05168	90.00001	2.57297
215.79126	90.00000	3.69775
227.21088	90.00000	4.53007
240.54947	90.00000	4.88665
254.84306	90.00000	4.50974
269.11914	90.00000	3.50245
282.42630	90.00000	2.25406
293.89075	90.00000	1.11273
302.28604	90.00000	0.43720
307.33142	90.00001	0.13964
310.00000	90.00000	0.00000
180.00002	70.00002	0.00000
181.87805	70.00001	1.60091
188.28549	70.00001	3.16227
198.61383	70.00002	4.55034
212.11540	70.00001	5.57775
227.88853	70.00001	6.01690
244.79076	70.00001	5.55048
261.66962	70.00002	4.30794
277.40085	70.00001	2.77049
290.95337	70.00001	1.36664
300.87878	70.00001	0.53662
306.84445	70.00001	0.17135
310.00000	70.00001	0.00000
160.00002	50.00002	0.00000
162.13443	50.00001	1.89290
169.51930	50.00002	3.75157
181.43639	50.00002	5.40293
197.01996	50.00002	6.62543
215.22762	50.00002	7.14715
234.73846	50.00002	6.59123
254.22009	50.00001	5.11344
272.37543	50.00002	3.28693
288.01602	50.00002	1.62056
299.[illegible]7150	50.00001	0.63604
306.35745	50.00001	0.20307
310.00000	50.00001	0.00000
140.00000	30.00000	0.00000
142.06560	29.67057	2.18971
150.13760	29.34405	4.36020

ORIGINAL PAGE IS
OF POOR QUALITY

```
          163.44543              29.05283               6.29589
          180.83961              28.56271               7.74840
          201.50874              28.32885               8.37185
          223.97173              28.57864               7.70594
          246.43689              29.10410               5.95502
          267.20563              29.42558               3.81819
          285.03714              29.71745               1.87806
          298.05640              29.88927               0.73601
          305.86963              29.96467               0.23484
          310.00000              30.00000               0.00000
TRIANGULAR PANEL TOLERANCE = .003
BOUNDARY CONDITION = 1,UPPER
*/
*/
 NETWORK=LWR-WING-8        13        7
          310.00000             150.00000               0.00000
          308.79236             150.00000              -0.04450
          306.50790             150.00000              -0.13894
          302.70288             150.00000              -0.35098
          297.50259             150.00000              -0.70477
          291.46768             150.00000              -1.08595
          285.00000             150.00000              -1.38750
          278.53229             150.00000              -1.49589
          272.47731             150.00000              -1.38703
          267.32361             150.00000              -1.13998
          263.35028             150.00000              -0.80508
          260.85251             150.00000              -0.43294
          260.00000             150.00000               0.00000
          310.00000             130.00000               0.00000
          308.30539             130.00000              -0.07621
          305.10062             130.00000              -0.23836
          299.76550             130.00000              -0.60489
          292.47714             130.00000              -1.22120
          284.01816             130.00000              -1.89145
          274.94769             130.00000              -2.42825
          265.87137             130.00000              -2.62614
          257.40186             130.00000              -2.43471
          250.14616             130.00000              -1.99257
          244.58409             130.00000              -1.39438
          241.10889             130.00000              -0.72493
          240.00000             130.00000               0.00000
          310.00000             110.00001               0.00000
          307.81842             110.00001              -0.10793
          303.69336             110.00002              -0.33778
          296.82813             110.00001              -0.85881
          287.45172             110.00001              -1.73763
          276.56866             110.00001              -2.69695
          264.89539             110.00001              -3.46899
          253.21045             110.00001              -3.75640
          242.30638             110.00001              -3.48239
          232.96873             110.00001              -2.84516
          225.81792             110.00001              -1.98368
          221.36528             110.00001              -1.01692
          220.00002             110.00002               0.00000
          310.00000              90.00000               0.00000
          307.33145              90.00001              -0.13964
          302.28607              90.00000              -0.43720
```

ORIGINAL PAGE IS
OF POOR QUALITY

293.89075	90.00000	-1.11273
282.42630	90.00000	-2.25406
269.11914	90.00000	-3.50245
254.84306	90.00000	-4.50974
240.54951	90.00000	-4.88665
227.21091	90.00000	-4.53007
215.79129	90.00000	-3.69775
207.05173	90.00000	-2.57298
201.62164	90.00000	-1.30892
200.00000	90.00000	0.00000
310.00000	70.00001	0.00000
306.84445	70.00001	-0.17135
300.87878	70.00001	-0.53662
290.95337	70.00001	-1.36664
277.40085	70.00001	-2.77049
261.66962	70.00002	-4.30795
244.79076	70.00001	-5.55048
227.88861	70.00001	-6.01670
212.11543	70.00001	-5.57775
178.61386	70.00001	-4.55034
188.28554	70.00001	-3.16228
181.87804	70.00001	-1.60091
180.00002	70.00002	0.00000
310.00000	50.00001	0.00000
306.35748	50.00001	-0.20307
299.47150	50.00001	-0.63604
288.01599	50.00002	-1.62056
272.37543	50.00002	-3.28692
254.22009	50.00001	-5.11344
234.73846	50.00002	-6.59123
215.22766	50.00001	-7.14716
197.01997	50.00001	-6.62543
181.43642	50.00001	-5.40293
169.51938	50.00002	-3.75138
162.13441	50.00001	-1.89290
160.00002	50.00002	0.00000
310.00000	30.00000	0.00000
305.86966	29.96467	-0.23484
298.05643	29.88927	-0.73601
285.03711	29.71745	-1.87806
267.20563	29.42558	-3.81819
246.43689	29.10410	-5.95502
223.97174	28.57865	-7.70594
201.50683	28.32884	-8.37185
180.83968	28.56272	-7.74840
163.44548	29.05283	-6.29590
150.13768	29.34405	-4.36021
142.06560	29.67058	-2.18970
140.00000	30.00000	0.00000

TRIANGULAR PANEL TOLERANCE = .003
BOUNDARY CONDITION = 1,UPPER
*/
*/
*/
*/
NETWORK=WING-CLOS-9 13 2

260.00000	150.00000	0.00000

```
            260.85254            150.00000            0.43294
            263.35025            150.00000            0.80508
            267.32361            150.00000            1.13998
            272.49731            150.00000            1.38703
            278.53229            150.00000            1.49589
            285.00000            150.00000            1.38750
            291.46768            150.00000            1.08595
            297.50259            150.00000            0.70477
            302.70285            150.00000            0.35098
            306.50790            150.00000            0.13894
            308.79236            150.00000            0.04450
            310.00000            150.00000            0.00000
            260.00000            150.00000            0.00000
            260.85251            150.00000           -0.43294
            263.35028            150.00000           -0.80508
            267.32361            150.00000           -1.13998
            272.49731            150.00000           -1.38703
            278.53229            150.00000           -1.49589
            285.00000            150.00000           -1.38750
            291.46768            150.00000           -1.08595
            297.50259            150.00000           -0.70477
            302.70288            150.00000           -0.35098
            306.50790            150.00000           -0.13894
            308.79236            150.00000           -0.04450
            310.00000            150.00000            0.00000
TRIANGULAR PANEL TOLERANCE = .003
BOUNDARY CONDITION = 1,LOWER
*/
*/
 NETWORK=EXIT-10        15        2
            360.00000              0.00000           30.00000
            360.00000             10.37696           28.13042
            360.00000             16.03367           25.85336
            360.00000             19.50493           22.95823
            360.00000             23.06456           19.04439
            360.00000             26.49111           13.27077
            360.00000             29.01616            6.53964
            360.00000             30.00000            0.00000
            360.00000             29.01616           -6.53964
            360.00000             26.49111          -13.27077
            360.00000             23.06455          -19.04440
            360.00000             19.50492          -22.95823
            360.00000             16.03366          -25.85337
            360.00000             10.37693          -28.13043
            360.00000              0.00000          -30.00000
            360.0                  0.0               0.0
            360.0                  0.0               0.0
            360.0                  0.0               0.0
            360.0                  0.0               0.0
            360.0                  0.0               0.0
            360.0                  0.0               0.0
            360.0                  0.0               0.0
            360.0                  0.0               0.0
            360.0                  0.0               0.0
            360.0                  0.0               0.0
            360.0                  0.0               0.0
            360.0                  0.0               0.0
```

```
          360.0                   0.0                      0.0
          360.0                   0.0                      0.0
          360.0                   0.0                      0.0
 TRIANGULAR PANEL TOLERANCE = .003
 BOUNDARY CONDITION = 4,    8,1    6,3
 SINGULARITY TYPES = SA, DA
 */
 */
 NETWORK=WING-WAKE-11          5      7
          310.00000             150.00000                  0.00000
          326.66669             150.00000                  0.00000
          343.33334             150.00000                  0.00000
          360.00000             150.00000                  0.00000
          600.0                 150.0                      0.0
          310.00000             130.00000                  0.00000
          326.66669             130.00000                  0.00000
          343.33334             130.00000                  0.00000
          360.00000             130.00000                  0.00000
          600.0                 130.0                      0.0
          310.00000             110.00001                  0.00000
          326.66669             110.00001                  0.00000
          343.33334             110.00001                  0.00000
          360.00000             110.00001                  0.00000
          600.0                 110.00001                  0.0
          310.00000              90.00000                  0.00000
          326.66669              90.00000                  0.00000
          343.33334              90.00000                  0.00000
          360.00000              90.00000                  0.00000
          600.0                  90.00                     0.0
          310.00000              70.00001                  0.00000
          326.66669              70.00001                  0.00000
          343.33334              70.00001                  0.00000
          360.00000              70.00001                  0.00000
          600.0                  70.00001                  0.0
          310.00000              50.00001                  0.00000
          326.66669              50.00001                  0.00000
          343.33334              50.00001                  0.00000
          360.00000              50.00001                  0.00000
          600.0                  50.00001                  0.0
          310.00000              30.00000                  0.00000
          326.66669              30.00000                  0.00000
          343.33334              30.00000                  0.00000
          360.00000              30.00000                  0.00000
          600.0                  30.0                      0.0
 WAKE FLOW PROPERTIES TAG
 BOUNDARY CONDITION=1,WAKE 1
 */
 */
 NETWORK=EXIT-UPR-WAKE-12          2    8
          360.00000               0.00000                 30.00000
          600.00000               0.00000                 30.00000
          360.00000              10.37693                 28.13043
          600.00000              10.37693                 28.13043
          360.00000              16.03366                 25.85337
          600.00000              16.03366                 25.85337
          360.00000              19.50492                 22.95823
          600.00000              19.50492                 22.95823
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
           360.00000                23.06455                19.04440
           600.00000                23.06455                19.04440
           360.00000                26.49111                13.27077
           600.00000                26.49111                13.27077
           360.00000                29.01616                 6.53964
           600.00000                29.01616                 6.53964
           360.00000                30.00000                 0.00000
           600.00000                30.00000                 0.00000
TRIANGULAR PANEL TOLERANCE = .003
BOUNDARY CONDITION = 1, WAKE 1
 NETWORK=EXIT-LWR-WAKE-13          2    8
           360.00000                30.00000                 0.00000
           600.00000                30.00000                 0.00000
           360.00000                29.01616                -6.53964
           600.00000                29.01616                -6.53964
           360.00000                26.49111               -13.27077
           600.00000                26.49111               -13.27077
           360.00000                23.06456               -19.04439
           600.00000                23.06456               -19.04439
           360.00000                19.50493               -22.95823
           600.00000                19.50493               -22.95823
           360.00000                16.03367               -25.85336
           600.00000                16.03367               -25.85336
           360.00000                10.37696               -28.13042
           600.00000                10.37696               -28.13042
           360.00000                 0.00000               -30.00000
           600.00000                 0.00000               -30.00000
TRIANGULAR PANEL TOLERANCE = .003
BOUNDARY CONDITION = 1, WAKE 1
*/
*/
*/
*/
*/
BEGIN GEOMETRIC EDGE MATCHING
ABUTMENT = UPR-WING-7,     2  ENTIRE-EDGE   +
         = UPR-BODY-4,    1  ENTIRE-EDGE
ABUTMENT = LWR-WING-8,    2  ENTIRE-EDGE  +
         = LWR-BODY-3,    3  ENTIRE-EDGE
*/
*/
BEGIN FLOW PROPERTIES DATA
SURFACE FLOW PROPERTIES = CASE-1        /SF1
NETWORK-IMAGES=1,REVERSE=2,REVERSE=3,REVERSE=4,REVERSE=5,REVERSE     +
             =6,REVERSE=7=8=9,REVERSE
POINTS = CENTER              /SF4
PRINTOUT = 1,2,8,13,16                /SF10A
DATA BASE = ALL          /SF11A
*/
*/
FORCES AND MOMENTS          /FM1
REFERENCE PARAMETERS = SR, 37500.0, BR, 150.0, CR, 140.0       /FM2
AXIS SYSTEM = RCS,0.,0.,0., WAS, 0.,0.,0.      /FM3
PRINTOUT =   NETWORK, CONFIGURATION         /FM5
DATA = NETWORK,CONFIGURATION
CASE = CASE-1          /FM7
NETWORK-IMAGES =1,REVERSE=2,REVERSE=3,REVERSE=4,REVERSE=5,REVERSE=6,REVE
```

```
SURFACE = UPPER      /FM12
*/
*/
BEGIN PRINT PLOT DATA     /PP1
GEOMETRY DATA             /PP2A
NETWORK = 1,2,3,4,5,6,7,8      /PP2B
POINT DATA                //PP3A
ARRAY = COLUMNS,CONTROL-POINTS     /PP3E
CONFIGURATION DATA        /PP4A
END PROBLEM DEFINITION
%%EOR
%%EOF
```

C.3.3 New PAN AIR Problem

The new PAN AIR problem created by the Viscous Simulation Module is presented in the following pages. This problem contains the boundary conditions specified to simulate the boundary layer.

ORIGINAL PAGE IS
OF POOR QUALITY

```
  TY RIP44.TMP
ANNHW,PN,T1100,CM154000,YD2,YL2,R3.
CCOUNT(FHASB,T5520)
ISPOSE,OUTPUT,ST ARSIXD,*PR.
OMMENT. RUN THE PRODUCTION VERSION OF PANAIR
OMMENT. STORED AS [12,1]RUNPANAIR.JCL ON CAPAIR......................
OMMENT.
OPYSP(INPUT,OUTPUT)
EWIND(INPUT)
ETNAME(FHASBGD)
OUNT(VSN=D0111D)
EQUEST,TAPE9,SN=FHASBGD,*PF.
EQUEST,TAPE10,SN=FHASBGD,*PF.
EQUEST,TAPE11,SN=FHASBGD,*PF.
UDIT(ID=FHAICB,AI=P)
OMMENT. PURGE ANY OLD DATASETS THAT MIGHT BE THERE
XIT,U.
OMMENT. ATTACH AND EXECUTE THE MEC MODULE
TTACH(MEC,ID=PANAIR)
EC.
UMMARY.
ETURN(MEC)
OMMENT. ATTACH AND EXECUTE THE DIP MODULE
TTACH(DIP,ID=PANAIR)
IP(INPUT,PL=100000)
UMMARY.
ETURN(DIP)
OMMENT. ATTACH AND EXECUTE THE DQG MODULE
TTACH(DQG,ID=PANAIR)
QG(PL=100000)
UMMARY.
ETURN(DQG)
OMMENT. ATTACH AND EXECUTE THE MAG MODULE
TTACH(MAG,ID=PANAIR)
AG.
UMMARY.
ETURN,MAG.
OMMENT. ATTACH AND EXECUTE THE RMS MODULE
TTACH(RMS,ID=PANAIR)
MS.
UMMARY.
ETURN(RMS)
OMMENT. ATTACH AND EXECUTE THE RHS MODULE
TTACH(RHS,ID=PANAIR)
HS.
UMMARY.
ETURN(RHS)
OMMENT. ATTACH AND EXECUTE THE MDG MODULE
TTACH(MDG,ID=PANAIR)
DG.
UMMARY.
ETURN(MDG)
OMMENT. ATTACH AND EXECUTE THE PDP MODULE
TTACH(PDP,ID=PANAIR)
DP(PL=40000)
UMMARY.
ETURN(PDP)
OMMENT. ATTACH AND EXECUTE THE CDP MODULE
TTACH(CDP,CDP,ID=PANAIR)
DP.
UMMARY.
ETURN(CDP)
OMMENT. ATTACH AND EXECUTE THE PPP MODULE
TTACH(PPP,ID=PANAIR)
PP
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
UMMARY.
ETURN(PPP)
UMMARY.
XIT,U.
EWIND(TAPE9,TAPE10,TAPE11)
OPYSBF(TAPE9,OUTPUT)
OPYSBF(TAPE10,OUTPUT)
OPYSBF(TAPE11,OUTPUT)
XIT,U.
EWIND(INPUT)
ATALOG,TAPE9,PPP9G,SN=FHASBGD,ID=FHAICB.
ATALOG,TAPE10,PPP10G,SN=FHASBGD,ID=FHAICB.
ATALOG,TAPE11,PPP11G,SN=FHASBGG,ID=FHAICB.
UDIT(ID=FHAICB,AI=P)
%EOR
PAN AIR / THIS IS THE FIRST CARD OF MEC INPUT FOR CDC 7600/SCOPE
YSTEM PANAIR FHASBGD AMES
ID =CHECKOUT OF PANAIR METHOD FOR VSTOL ANALYSIS
ATA BASE DIRECTIVE BLOCK
APPEND DAR TO ALL
UID = FHAICB FOR ALL
SET = FHASBGD FOR ALL
MUID= PANAIR FOR ALL
MSET= FHASBGD FOR ALL
END DATA BASE DIRECTIVE BLOCK
HECK DATA RUN
XECUTION DIRECTIVE BLOCK
FIND POTENTIAL FLOW
END EXECUTION DIRECTIVE BLOCK
ND PAN AIR DIRECTIVES
%EOR
/
/
/
/ DIP INPUT - FILE RIP4.JOB
/   WING-BODY PROBLEM FOR CHECKOUT OF VISCOUS EFFECTS MODULE
/
/
/
/
EGIN GLOBAL DATA
PID=MODEL TO CHECK OUT VISCOUS EFFECTS MODULE
ID = BHATELEY/HOWELL GENERAL DYNAMICS 817 732 4811
ONFIGURATION = FIRST        /ONE PLANES OF SYMMETRY.
ACH = 0.600, CALPHA = 4.0, CBETA = 0.0/ RECORD G5
LPHA = 4.0
ID =FOUR-DEGREE                              /G6
OLERANCE = 0.1                                       /G7
URFACE SELECTION = UPPER                             /G8
ELECTION OF VELOCITY COMPUTATION= BOUNDARY-CONDITION  /G9
RESSURE COEFFICIENT RULES=ISENTROPIC,SECOND-ORDER      /G12
/
HECKOUT PRINTS = DIP,1,2,3,    DQG,1,4,5
EGIN NETWORK DATA
/  THESE ARE THE NETWORKS FOR WING-BODY MODEL
/
NETWORK=NOSE-1   15    4
           0.00000             0.00000             0.00000
           0.00000             0.00000             0.00000
           0.00000             0.00000             0.00000
           0.00000             0.00000             0.00000
           0.00000             0.00000             0.00000
           0.00000             0.00000             0.00000
           0.00000             0.00000             0.00000
           0.00000             0.00000             0.00000
           0.00000             0.00000             0.00000
```

```
       0.00000            0.00000            0.00000
       0.00000            0.00000            0.00000
       0.00000            0.00000            0.00000
       0.00000            0.00000            0.00000
       0.00000            0.00000            0.00000
       0.00000            0.00000            0.00000
       0.00000            0.00000            0.00000
      33.33334            0.00000          -10.00000
      33.33334            3.45899           -9.37681
      33.33334            5.34456           -8.61779
      33.33334            6.50164           -7.65274
      33.33334            7.68819           -6.34813
      33.33334            8.83037           -4.42359
      33.33334            9.67205           -2.17988
      33.33334           10.00000            0.00000
      33.33334            9.67205            2.17988
      33.33334            8.83037            4.42359
      33.33334            7.68818            6.34813
      33.33334            6.50164            7.65275
      33.33334            5.34455            8.61779
      33.33334            3.45898            9.37681
      33.33334            0.00000           10.00000
      66.66667            0.00000          -20.00000
      66.66667            6.91797          -18.75362
      66.66667           10.68912          -17.23558
      66.66667           13.00329          -15.30549
      66.66667           15.37637          -12.69626
      66.66667           17.66074           -8.84718
      66.66667           19.34411           -4.35976
      66.66667           20.00000            0.00000
      66.66667           19.34411            4.35976
      66.66667           17.66074            8.84718
      66.66667           15.37637           12.69627
      66.66667           13.00328           15.30549
      66.66667           10.68910           17.23558
      66.66667            6.91795           18.75362
      66.66667            0.00000           20.00000
     100.00000            0.00000          -30.00000
     100.00000           10.37696          -28.13042
     100.00000           16.03367          -25.85336
     100.00000           19.50493          -22.95823
     100.00000           23.06456          -19.04439
     100.00000           26.49111          -13.27077
     100.00000           29.01616           -6.53964
     100.00000           30.00000            0.00000
     100.00000           29.01616            6.53964
     100.00000           26.49111           13.27077
     100.00000           23.06455           19.04440
     100.00000           19.50492           22.95823
     100.00000           16.03366           25.85337
     100.00000           10.37693           28.13043
     100.00000            0.00000           30.00000
RIANGULAR PANEL TOLERANCE = .003
OUNDARY CONDITION = 1,LOWER
/
/
NETWORK=FWD-BODY-2       15       3
     100.00000            0.00000          -30.00000
     100.00000           10.37696          -28.13042
     100.00000           16.03367          -25.85336
     100.00000           19.50493          -22.95823
     100.00000           23.06456          -19.04439
     100.00000           26.49111          -13.27077
     100.00000           29.01616           -6.53964
     100.00000           30.00000            0.00000
     100.00000           29.01616            6.53964
```

```
         100.00000            26.49111            13.27077
         100.00000            23.06455            19.04440
         100.00000            19.50492            22.95823
         100.00000            16.03366            25.85337
         100.00000            10.37693            28.13043
         100.00000             0.00000            30.00000
         120.00000             0.00000           -30.00000
         120.00000            10.37696           -28.13042
         120.00000            16.03367           -25.85336
         120.00000            19.50493           -22.95823
         120.00000            23.06456           -19.04439
         120.00000            26.49111           -13.27077
         120.00000            29.01616            -6.53964
         120.00000            30.00000             0.00000
         120.00000            29.01616             6.53964
         120.00000            26.49111            13.27077
         120.00000            23.06455            19.04440
         120.00000            19.50492            22.95823
         120.00000            16.03366            25.85337
         120.00000            10.37693            28.13043
         120.00000             0.00000            30.00000
         140.00000             0.00000           -30.00000
         140.00000            10.37696           -28.13042
         140.00000            16.03367           -25.85336
         140.00000            19.50493           -22.95823
         140.00000            23.06456           -19.04439
         140.00000            26.49111           -13.27077
         140.00000            29.01616            -6.53964
         140.00000            30.00000             0.00000
         140.00000            29.01616             6.53964
         140.00000            26.49111            13.27077
         140.00000            23.06455            19.04440
         140.00000            19.50492            22.95823
         140.00000            16.03366            25.85337
         140.00000            10.37693            28.13043
         140.00000             0.00000            30.00000
RIANGULAR PANEL TOLERANCE = .003
OUNDARY CONDITION = 1,LOWER
/
/
NETWORK=LWR-BODY-3      8     7
         140.00000             0.00000           -30.00000
         140.00000            10.37696           -28.13042
         140.00000            16.03367           -25.85336
         140.00000            19.50493           -22.95823
         140.00000            23.06456           -19.04439
         140.00000            26.49111           -13.27077
         140.00000            29.01616            -6.53964
         140.00000            30.00000             0.00000
         150.13768             0.00000           -30.00000
         150.13768             9.43575           -28.40252
         150.13768            14.67512           -26.57329
         150.13768            18.03844           -24.32530
         150.13768            21.47990           -20.95077
         150.13768            25.04300           -16.03498
         150.13768            27.85628           -10.14802
         150.13768            29.34405            -4.36021
         180.83968             0.00000           -30.00000
         180.83968             8.68515           -28.60125
         180.83968            13.52987           -27.06412
         180.83968            16.81413           -25.32869
         180.83968            20.10723           -22.37667
         180.83968            23.72057           -18.13178
         180.83968            26.65920           -12.91335
         180.83968            28.56272            -7.74840
         223.97174             0.00000           -30.00000
         223.97174             8.67491           -28.59976
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
          223.97174           13.54501           -27.05814
          223.97174           16.83045           -25.31665
          223.97174           20.12545           -22.35871
          223.97174           23.73869           -18.10545
          223.97174           26.67559           -12.87822
          223.97174           28.37865            -7.70594
          267.20563            0.00000           -30.00000
          267.20563            9.55310           -28.37002
          267.20563           14.85005           -26.49034
          267.20563           18.22490           -24.15783
          267.20563           21.68755           -20.71991
          267.20563           25.23666           -15.69620
          267.20563           28.02673            -9.70523
          267.20563           29.42558            -3.81819
          298.05643            0.00000           -30.00000
          298.05643           10.21853           -28.17808
          298.05643           15.81293           -25.98376
          298.05643           19.26217           -23.18852
          298.05643           22.81251           -19.37369
          298.05643           26.26452           -13.74125
          298.05643           28.85916            -7.15470
          298.05643           29.88927            -0.73601
          310.00000            0.00000           -30.00000
          310.00000           10.37696           -28.13042
          310.00000           16.03367           -25.85336
          310.00000           19.50493           -22.95823
          310.00000           23.06456           -19.04439
          310.00000           26.49111           -13.27077
          310.00000           29.01616            -6.53964
          310.00000           30.00000             0.00000
RIANGULAR PANEL TOLERANCE = .003
OUNDARY CONDITION = 1,LOWER
/
/
NETWORK=UPR-BODY-4     8     7
          140.00000           30.00000             0.00000
          140.00000           29.01616             6.53964
          140.00000           26.49111            13.27077
          140.00000           23.06455            19.04440
          140.00000           19.50492            22.95823
          140.00000           16.03366            25.85337
          140.00000           10.37693            28.13043
          140.00000            0.00000            30.00000
          150.13760           29.34405             4.36020
          150.13760           27.85629            10.14801
          150.13760           25.04301            16.03498
          150.13760           21.47990            20.95077
          150.13760           18.03843            24.32530
          150.13760           14.67511            26.57329
          150.13760            9.43572            28.40253
          150.13760            0.00000            30.00000
          180.83961           28.56271             7.74840
          180.83961           26.65920            12.91336
          180.83961           23.72057            18.13179
          180.83961           20.10723            22.37668
          180.83961           16.81412            25.32869
          180.83961           13.52985            27.06412
          180.83961            8.68512            28.60125
          180.83961            0.00000            30.00000
          223.97173           28.57864             7.70594
          223.97173           26.67559            12.87823
          223.97173           23.73868            18.10546
          223.97173           20.12545            22.35872
          223.97173           16.83044            25.31665
          223.97173           13.54499            27.05814
```

ORIGINAL PAGE IS
OF POOR QUALITY

223.97173	8.69489	28.59877
223.97173	0.00000	30.00000
267.20563	29.42558	3.81819
267.20563	28.02672	9.70524
267.20563	25.23666	15.69621
267.20563	21.68754	20.71992
267.20563	18.22488	24.15784
267.20563	14.85002	26.49035
267.20563	9.55307	28.37003
267.20563	0.00000	30.00000
298.05640	29.88927	0.73601
298.05640	28.85916	7.15471
298.05640	26.26452	13.74126
298.05640	22.81250	19.37370
298.05640	19.26215	23.18853
298.05640	15.81291	25.98376
298.05640	10.21850	28.17809
298.05640	0.00000	30.00000
310.00000	30.00000	0.00000
310.00000	29.01616	6.53964
310.00000	26.49111	13.27077
310.00000	23.06455	19.04440
310.00000	19.50492	22.95823
310.00000	16.03366	25.85337
310.00000	10.37693	28.13043
310.00000	0.00000	30.00000

RIANGULAR PANEL TOLERANCE = .003
OUNDARY CONDITION = 1,LOWER
/
/
NETWORK=LWR-AFT-BODY-5 8 4

310.00000	0.00000	-30.00000
310.00000	10.37696	-28.13042
310.00000	16.03367	-25.85336
310.00000	19.50493	-22.95823
310.00000	23.06456	-19.04439
310.00000	26.49111	-13.27077
310.00000	29.01616	-6.53964
310.00000	30.00000	0.00000
326.66669	0.00000	-30.00000
326.66669	10.37696	-28.13042
326.66669	16.03367	-25.85336
326.66669	19.50493	-22.95823
326.66669	23.06456	-19.04439
326.66669	26.49111	-13.27077
326.66669	29.01616	-6.53964
326.66669	30.00000	0.00000
343.33334	0.00000	-30.00000
343.33334	10.37696	-28.13042
343.33334	16.03367	-25.85336
343.33334	19.50493	-22.95823
343.33334	23.06456	-19.04439
343.33334	26.49111	-13.27077
343.33334	29.01616	-6.53964
343.33334	30.00000	0.00000
360.00000	0.00000	-30.00000
360.00000	10.37696	-28.13042
360.00000	16.03367	-25.85336
360.00000	19.50493	-22.95823
360.00000	23.06456	-19.04439
360.00000	26.49111	-13.27077
360.00000	29.01616	-6.53964
360.00000	30.00000	0.00000

RIANGULAR PANEL TOLERANCE = .003
OUNDARY CONDITION = 1,LOWER
/
/

ORIGINAL PAGE IS
OF POOR QUALITY

NETWORK=UPR-AFT-BODY-6 8 4

310.00000	30.00000	0.00000
310.00000	29.01616	6.53964
310.00000	26.49111	13.27077
310.00000	23.06455	19.04440
310.00000	19.50492	22.95823
310.00000	16.03366	25.85337
310.00000	10.37693	28.13043
310.00000	0.00000	30.00000
326.66669	30.00000	0.00000
326.66669	29.01616	6.53964
326.66669	26.49111	13.27077
326.66669	23.06455	19.04440
326.66669	19.50492	22.95823
326.66669	16.03366	25.85337
326.66669	10.37693	28.13043
326.66669	0.00000	30.00000
343.33334	30.00000	0.00000
343.33334	29.01616	6.53964
343.33334	26.49111	13.27077
343.33334	23.06455	19.04440
343.33334	19.50492	22.95823
343.33334	16.03366	25.85337
343.33334	10.37693	28.13043
343.33334	0.00000	30.00000
360.00000	30.00000	0.00000
360.00000	29.01616	6.53964
360.00000	26.49111	13.27077
360.00000	23.06455	19.04440
360.00000	19.50492	22.95823
360.00000	16.03366	25.85337
360.00000	10.37693	28.13043
360.00000	0.00000	30.00000

RIANGULAR PANEL TOLERANCE = .003
OUNDARY CONDITION = [illegible]OWER
/
/
NETWORK=UPR-WING-7 13 7

260.00000	150.00000	0.00000
260.85254	150.00000	0.43294
263.35025	150.00000	0.80508
267.32361	150.00000	1.13998
272.49731	150.00000	1.38703
278.53229	150.00000	1.49589
285.00000	150.00000	1.38750
291.46768	150.00000	1.08595
297.50259	150.00000	0.70477
302.70285	150.00000	0.35098
306.50790	150.00000	0.13894
308.79236	150.00000	0.04450
310.00000	150.00000	0.00000
240.00000	130.00000	0.00000
241.10892	130.00000	0.72493
244.58406	130.00000	1.39437
250.14616	130.00000	1.99257
257.40182	130.00000	2.43471
265.87134	130.00000	2.62614
274.94769	130.00000	2.42825
284.01816	130.00000	1.89145
292.47714	130.00000	1.22120
299.76547	130.00000	0.60489
305.10062	130.00000	0.23836
308.30539	130.00000	0.07621
310.00000	130.00000	0.00000
220.00002	110.00002	0.00000
[illegible]	[illegible]	[illegible]

ORIGINAL PAGE IS
OF POOR QUALITY

221.38331	110.00002	1.01893
225.81787	110.00001	1.98367
232.96872	110.00001	2.84516
242.30637	110.00001	3.48239
253.21042	110.00001	3.75640
264.89539	110.00001	3.46899
276.56866	110.00001	2.69695
287.45172	110.00001	1.73763
296.82813	110.00001	0.85881
303.69333	110.00002	0.33778
307.81839	110.00001	0.10793
310.00000	110.00001	0.00000
200.00000	90.00000	0.00000
201.62166	90.00000	1.30892
207.05168	90.00001	2.57297
215.79126	90.00000	3.69775
227.21088	90.00000	4.53007
240.54947	90.00000	4.88665
254.84306	90.00000	4.50974
269.11914	90.00000	3.50245
282.42630	90.00000	2.25406
293.89075	90.00000	1.11273
302.28604	90.00000	0.43720
307.33142	90.00001	0.13964
310.00000	90.00000	0.00000
180.00002	70.00002	0.00000
181.87805	70.00001	1.60091
188.28549	70.00001	3.16227
198.61383	70.00002	4.55034
212.11540	70.00001	5.57775
227.88853	70.00001	6.01690
244.79076	70.00001	5.55048
261.66962	70.00002	4.30794
277.40085	70.00001	2.77049
290.95337	70.00001	1.36664
300.87878	70.00001	0.53662
306.84445	70.00001	0.17135
310.00000	70.00001	0.00000
160.00002	50.00002	0.00000
162.13443	50.00001	1.89290
169.51930	50.00002	3.75157
181.43639	50.00002	5.40293
197.01996	50.00002	6.62543
215.22762	50.00002	7.14715
234.73846	50.00002	6.59123
254.22009	50.00001	5.11344
272.37543	50.00002	3.28693
288.01602	50.00002	1.62056
299.47150	50.00001	0.63604
306.35745	50.00001	0.20307
310.00000	50.00001	0.00000
140.00000	30.00000	0.00000
142.06560	29.67057	2.18971
150.13760	29.34405	4.36020
163.44543	29.05283	6.29589
180.83961	28.56271	7.74840
201.50874	28.32885	8.37185
223.97173	28.57864	7.70594
246.43689	29.10410	5.95502
267.20563	29.42558	3.81819
285.03714	29.71745	1.87806
298.05640	29.88927	0.73601
305.86963	29.96467	0.23484
310.00000	30.00000	0.00000

RIANGULAR PANEL TOLERANCE = .003

/

```
OUNDARY CONDITION = 2,UPPE
PECIFIED FLOW
ERM = 1
OINTS = CENTER
/  OUTFLOW VELOCITIES FOR PANEL CENTER POINTS
 0.005660    0.002210    0.002249    0.001687    0.001465    0.001469
 0.001730    0.002063    0.002493    0.002812    0.002835    0.006688
 0.003700    0.001813    0.001980    0.001515    0.001357    0.001444
 0.001795    0.002244    0.002847    0.003271    0.002844    0.003836
 0.002754    0.001596    0.001811    0.001409    0.001260    0.001346
 0.001704    0.002181    0.002851    0.003406    0.003157    0.006016
 0.002209    0.001455    0.001687    0.001329    0.001187    0.001265
 0.001606    0.002094    0.002750    0.003172    0.002996    0.005717
 0.001857    0.001352    0.001598    0.001265    0.001118    0.001186
 0.001502    0.001966    0.002597    0.002988    0.002575    0.003155
 0.001618    0.001291    0.001531    0.001216    0.001008    0.001001
 0.001334    0.001879    0.002473    0.002720    0.002700    0.004419
/
/
/
OINTS = EDGE
/  BEGIN VELOCITIES FOR EDGE 1
     0.006597
     0.004185
     0.003042
     0.002391
     0.001977
     0.001692
/
/
/  BEGIN EDGE VELOCITIES FOR SIDE 2
     0.001497
     0.001259
     0.001497
     0.001191
     0.000950
     0.000904
     0.001247
     0.001835
     0.002410
     0.002585
     0.002763
     0.005055
/
/
/  BEGIN EDGE VELOCITIES FOR SIDE 3
     0.005011
     0.003355
     0.006656
     0.007007
     0.004176
     0.010704
/
/
/  BEGIN EDGE VELOCITIES FOR SIDE 4
     0.011102
     0.002831
     0.002583
     0.002317
     0.001972
     0.001697
     0.001481
     0.001519
     0.001772
     0.002383
     [illegible]
```

```
     0.006636
/
/
DINIS = ADDITIONAL
 ( 1 ) =      0.003314
 ( 2 ) =      0.001933
 ( 3 ) =      0.006152
 ( 4 ) =      0.003225
/
/
NETWORK=LWR-WING-8       13        7
           310.00000          150.00000              0.00000
           308.79236          150.00000             -0.04450
           306.50790          150.00000             -0.13894
           302.70288          150.00000             -0.35098
           297.50259          150.00000             -0.70477
           291.46768          150.00000             -1.08595
           285.00000          150.00000             -1.38750
           278.53229          150.00000             -1.49589
           272.49731          150.00000             -1.38703
           267.32361          150.00000             -1.13998
           263.35028          150.00000             -0.80508
           260.85251          150.00000             -0.43294
           260.00000          150.00000              0.00000
           310.00000          130.00000              0.00000
           308.30539          130.00000             -0.07621
           305.10062          130.00000             -0.23836
           299.76550          130.00000             -0.60489
           292.47714          130.00000             -1.22120
           284.01816          130.00000             -1.89145
           274.94769          130.00000             -2.42825
           265.87137          130.00000             -2.62614
           257.40186          130.00000             -2.43471
           250.14616          130.00000             -1.99257
           244.58409          130.00000             -1.39438
           241.10889          130.00000             -0.72493
           240.00000          130.00000              0.00000
           310.00000          110.00001              0.00000
           307.81842          110.00001             -0.10793
           303.69336          110.00002             -0.33778
           296.82813          110.00001             -0.85881
           287.45172          110.00001             -1.73763
           276.56866          110.00001             -2.69695
           264.89539          110.00001             -3.46899
           253.21045          110.00001             -3.75640
           242.30638          110.00001             -3.48239
           232.96873          110.00001             -2.84516
           225.81792          110.00001             -1.98368
           221.36528          110.00001             -1.01692
           220.00002          110.00002              0.00000
           310.00000           90.00000              0.00000
           307.33145           90.00001             -0.13964
           302.28607           90.00000             -0.43720
           293.89075           90.00000             -1.11273
           282.42630           90.00000             -2.25406
           269.11914           90.00000             -3.50245
           254.84306           90.00000             -4.50974
           240.54951           90.00000             -4.88665
           227.21091           90.00000             -4.53007
           215.79129           90.00000             -3.69775
           207.05173           90.00000             -2.57298
           201.62164           90.00000             -1.30892
           200.00000           90.00000              0.00000
           310.00000           70.00001              0.00000
           306.84445           70.00001             -0.17135
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
          [illegible]              [illegible]              [illegible]
          290.95337                70.00001                 -1.36664
          277.40085                70.00001                 -2.77049
          261.66962                70.00002                 -4.30795
          244.79076                70.00001                 -5.55048
          227.88861                70.00001                 -6.01690
          212.11543                70.00001                 -5.57775
          198.61386                70.00001                 -4.55034
          188.28554                70.00001                 -3.16228
          181.87804                70.00001                 -1.60091
          180.00002                70.00002                  0.00000
          310.00000                50.00001                  0.00000
          306.35748                50.00001                 -0.20307
          299.47150                50.00001                 -0.63604
          288.01599               .50.00002                 -1.62056
          272.37543                50.00002                 -3.28692
          254.22009                50.00001                 -5.11344
          234.73846                50.00002                 -6.59123
          215.22766                50.00001                 -7.14716
          197.01997                50.00001                 -6.62543
          181.43642                50.00001                 -5.40293
          169.51938                50.00002                 -3.75158
          162.13441                50.00001                 -1.89290
          160.00002                50.00002                  0.00000
          310.00000                30.00000                  0.00000
          305.86966                29.96467                 -0.23484
          298.05543                29.88927                 -0.73601
          285.03711                29.71745                 -1.87806
          267.20563                29.42558                 -3.81819
          246.43689                29.10410                 -5.95502
          223.97174                28.57865                 -7.70594
          201.50883                28.32884                 -8.37185
          180.83968                28.56272                 -7.74840
          163.44548                29.05283                 -6.29590
          150.13768                29.34405                 -4.36021
          142.06560                29.67058                 -2.18970
          140.00000                30.00000                  0.00000
RIANGULAR PANEL TOLERANCE = .003
/
/
/
/
OUNDARY CONDITION = 2,UPPE
PECIFIED FLOW
ERM.= 1
OINTS = CENTER
/  OUTFLOW VELOCITIES FOR PANEL CENTER POINTS
 0.000077    0.001429    0.001891    0.001732    0.001504    0.001318
 0.001163    0.001108    0.001114    0.001043    0.001785    0.000000
 0.002341    0.002049    0.002299    0.001957    0.001570    0.001273
 0.001051    0.000995    0.001053    0.001085    0.001366    0.000000
 0.002164    0.001961    0.002357    0.001987    0.001561    0.001246
 0.001023    0.000978    0.001045    0.001127    0.001084    0.000000
 0.002722    0.002011    0.002283    0.001976    0.001548    0.001219
 0.001002    0.000963    0.001035    0.001155    0.000897    0.000000
 0.004242    0.002297    0.002248    0.001906    0.001492    0.001175
 0.000983    0.000954    0.001032    0.001173    0.000764    0.000000
 0.002936    0.002140    0.002176    0.001721    0.001291    0.001429
 0.001037    0.000831    0.001007    0.001149    0.000691    0.000000
/
/
/
OINTS = EDGE
/  BEGIN VELOCITIES FOR EDGE 1
     0.000000
     0.002442
     [illegible]
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
     0.002255
     0.002968
     0.004918
     0.003211
/
/
/   BEGIN EDGE VELOCITIES FOR SIDE 2
     0.002279
     0.002060
     0.002139
     0.001627
     0.001188
     0.001560
     0.001065
     0.000766
     0.000993
     0.001137
     0.000654
     0.000000
/
/
/   BEGIN EDGE VELOCITIES FOR SIDE 3
     0.000000
     0.000000
     0.000000
     0.000000
     0.000000
     0.000000
/
/
/   BEGIN EDGE VELOCITIES FOR SIDE 4
     0.000000
     0.001993
     0.001022
     0.001144
     0.001164
     0.001219
     0.001341
     0.001471
     0.001620
     0.001687
     0.001120
     0.000000
/
/
OINTS = ADDITIONAL
 ( 1 ) =        0.002629
 ( 2 ) =        0.001289
 ( 3 ) =        0.000000
 ( 4 ) =        0.001149
/
/
NETWORK=WING-CLOS-9        13        2
          260.00000          150.00000             0.00000
          260.85254          150.00000             0.43294
          263.35025          150.00000             0.80508
          267.32361          150.00000             1.13998
          272.49731          150.00000             1.38703
          278.53229          150.00000             1.49589
          285.00000          150.00000             1.38750
          291.46768          150.00000             1.08595
          297.50259          150.00000             0.70477
          302.70285          150.00000             0.35098
          306.50790          150.00000             0.13894
          308.79236          150.00000             0.04450
          310.00000          150.00000             0.00000
          260.00000          150.00000             0.00000
```

```
          260.85251           150.00000            -0.43294
          263.35028           150.00000            -0.80508
          267.32361           150.00000            -1.13998
          272.49731           150.00000            -1.38703
          278.53229           150.00000            -1.49589
          285.00000           150.00000            -1.38750
          291.46768           150.00000            -1.08595
          297.50259           150.00000            -0.70477
          302.70288           150.00000            -0.35098
          306.50790           150.00000            -0.13894
          308.79236           150.00000            -0.04450
          310.00000           150.00000             0.00000
RIANGULAR PANEL TOLERANCE = .003
OUNDARY CONDITION - 1,LOWER
/
/
NETWORK=EXIT-10       15        2
          360.00000             0.00000            30.00000
          360.00000            10.37696            28.13042
          360.00000            16.03367            25.85336
          360.00000            19.50493            22.95823
          360.00000            23.06456            19.04439
          360.00000            26.49111            13.27077
          360.00000            29.01616             6.53964
          360.00000            30.00000             0.00000
          360.00000            29.01616            -6.53964
          360.00000            26.49111           -13.27077
          360.00000            23.06455           -19.04440
          360.00000            19.50492           -22.95823
          360.00000            16.03366           -25.85337
          360.00000            10.37693           -28.13043
          360.00000             0.00000           -30.00000
          360.0                 0.0                0.0
          360.0                 0.0                0.0
          360.0                 0.0                0.0
          360.0                 0.0                0.0
          360.0                 0.0                0.0
          360.0                 0.0                0.0
          360.0                 0.0                0.0
          360.0                 0.0                0.0
          360.0                 0.0                0.0
          360.0                 0.0                0.0
          360.0                 0.0                0.0
          360.0                 0.0                0.0
          360.0                 0.0                0.0
          360.0                 0.0                0.0
          360.0                 0.0                0.0
RIANGULAR PANEL TOLERANCE = .003
OUNDARY CONDITION = 4,    8,1     6,3
INGULARITY TYPES = SA, DA
/
/
ETWORK=WING-WAKE-11        5        7
          310.00000           150.00000             0.00000
          326.66669           150.00000             0.00000
          343.33334           150.00000             0.00000
          360.00000           150.00000             0.00000
          600.0               150.0                 0.0
          310.00000           130.00000             0.00000
          326.66669           130.00000             0.00000
          343.33334           130.00000             0.00000
          360.00000           130.00000             0.00000
          600.0               130.0                 0.0
          310.00000           110.00001             0.00000
          326.66669           110.00001             0.00000
          343.33334           110.00001             0.00000
```

```
         360.00000          110.00001          0.00000
         600.0              110.00001          0.0
         310.00000           90.00000          0.00000
         326.66669           90.00000          0.00000
         343.33334           90.00000          0.00000
         360.00000           90.00000          0.00000
         600.0               90.00             0.0
         310.00000           70.00001          0.00000
         326.66669           70.00001          0.00000
         343.33334           70.00001          0.00000
         360.00000           70.00001          0.00000
         600.0               70.00001          0.0
         310.00000           50.00001          0.00000
         326.66669           50.00001          0.00000
         343.33334           50.00001          0.00000
         360.00000           50.00001          0.00000
         600.0               50.00001          0.0
         310.00000           30.00000          0.00000
         326.66669           30.00000          0.00000
         343.33334           30.00000          0.00000
         360.00000           30.00000          0.00000
         600.0               30.0              0.0
AKE FLOW PROPERTIES TAG
OUNDARY CONDITION=1,WAKE 1
/
/
NETWORK=EXIT-UPR-WAKE-12           2    8
         360.00000           0.00000          30.00000
         600.00000           0.00000          30.00000
         360.00000          10.37693          28.13043
         600.00000          10.37693          28.13043
         360.00000          16.03366          25.85337
         600.00000          16.03366          25.85337
         360.00000          19.50492          22.95823
         600.00000          19.50492          22.95823
         360.00000          23.06455          19.04440
         600.00000          23.06455          19.04440
         360.00000          26.49111          13.27077
         600.00000          26.49111          13.27077
         360.00000          29.01616           6.53964
         600.00000          29.01616           6.53964
         360.00000          30.00000           0.00000
         600.00000          30.00000           0.00000
RIANGULAR PANEL TOLERANCE = .003
OUNDARY CONDITION = 1, WAKE 1
NETWORK=EXIT-LWR-WAKE-13           2    8
         360.00000          30.00000           0.00000
         600.00000          30.00000           0.00000
         360.00000          29.01616          -6.53964
         600.00000          29.01616          -6.53964
         360.00000          26.49111         -13.27077
         600.00000          26.49111         -13.27077
         360.00000          23.06456         -19.04439
         600.00000          23.06456         -19.04439
         360.00000          19.50493         -22.95823
         600.00000          19.50493         -22.95823
         360.00000          16.03367         -25.85336
         600.00000          16.03367         -25.85336
         360.00000          10.37696         -28.13042
         600.00000          10.37696         -28.13042
         360.00000           0.00000         -30.00000
         600.00000           0.00000         -30.00000
RIANGULAR PANEL TOLERANCE = .003
OUNDARY CONDITION = 1, WAKE 1
/
/
```

ORIGINAL PAGE IS OF POOR QUALITY

```
/
/
/
EGIN GEOMETRIC EDGE MATCHING
BUTMENT = UPR-WING-7,     2  ENTIRE-EDGE    +
        = UPR-BODY-4,    1  ENTIRE-EDGE
BUTMENT = LWR-WING-8,    2  ENTIRE-EDGE  +
        = LWR-BODY-3,    3  ENTIRE-EDGE
/
/
EGIN FLOW PROPERTIES DATA
URFACE FLOW PROPERTIES = CASE-1        /SF1
ETWORK-IMAGES=1,REVERSE=2,REVERSE=3,REVERSE=4,REVERSE=5,REVERSE     +
             =6,REVERSE=7=8=9,REVERSE
OINTS = CENTER              /SF4
RINTOUT = 1,2,8,13,16                  /SF10A
/
/
ORCES AND MOMENTS         /FM1
EFERENCE PARAMETERS = SR, 37500.0, BR, 150.0, CR, 140.0      /FM2
XIS SYSTEM = RCS,0.,0.,0., WAS, 0.,0.,0.      /FM3
RINTOUT =   NETWORK, CONFIGURATION         /FM5
ATA = NETWORK,CONFIGURATION
ASE = CASE-1          /FM7
ETWORK-IMAGES =1,REVERSE=2,REVERSE=3,REVERSE=4,REVERSE=5,REVERSE=6,REVERSE=7=8
URFACE = UPPER      /FM12
/
/
EGIN PRINT PLOT DATA     /PP1
EOMETRY DATA             /PP2A
ETWORK = 1,2,3,4,5,6,7,8       /PP2B
OINT DATA                //PP3A
RRAY = COLUMNS,CONTROL-POINTS     /PP3E
ONFIGURATION DATA        /PP4A
ND PROBLEM DEFINITION
%EOR
%EOF
%EOF
$
```